

Express Mail No. EM449338449 U.S. PTO

PATENT

25484.0058



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Yoshimasa ISOZAKI, )  
Hideyuki MASUDA, Hideo SUZUKI, )  
Masahiro SHIMIZU, and Masashi HIRANO )  
Serial No.: Currently unknown )  
Filing Date: Concurrently herewith )  
For: SOUND SOURCE SYSTEM BASED ON )  
COMPUTER SOFTWARE AND METHOD OF )  
GENERATING ACOUSTIC WAVEFORM DATA )

#3  
R.T.  
1/25/98

TRANSMITTAL OF PRIORITY DOCUMENTS

Assistant Commissioner for Patents  
Washington, D.C. 20231

Dear Sir:

Enclosed herewith are certified copies of Japanese Patent Application No. 8-246942 filed August 30, 1996, Japanese Patent Application No. 8-248592 filed August 30, 1996 and Japanese Patent Application No. 9-017333 filed on January 14, 1997 from which priority is claimed under 35 U.S.C. 119 and Rule 55b.

Acknowledgment of these priority documents is respectfully requested to ensure that the subject information appears on the printed patent.

Respectfully submitted,

Brian M. Berliner  
Attorney for Applicant  
Reg. No. 34,549

Date: August 29, 1997

GRAHAM & JAMES LLP  
801 S. Figueroa St., 14th Floor  
Los Angeles, California 90017-5554  
Telephone: (213) 624-2500

日 本 国 特 許 庁

PATENT OFFICE  
JAPANESE GOVERNMENT



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

1 9 9 6 年 8 月 3 0 日

出 願 番 号

Application Number:

平成 8 年特許願第 2 4 6 9 4 2 号

出 願 人

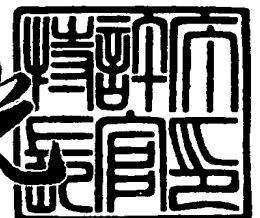
Applicant (s):

ヤマハ株式会社

1 9 9 7 年 6 月 1 3 日

特 許 庁 長 官  
Commissioner,  
Patent Office

荒井 寿光



出証番号 出証特平 0 9 - 3 0 4 5 6 9 2

【書類名】 特許願

【整理番号】 C25954

【提出日】 平成 8年 8月30日

【あて先】 特許庁長官殿

【国際特許分類】 G10H 7/00  
G06F 9/00

【発明の名称】 コンピュータソフトウェアを用いた音源システムおよび  
音響波形データ生成方法

【請求項の数】 11

【発明者】

    【住所又は居所】 静岡県浜松市中沢町10番1号 ヤマハ株式会社内

    【氏名】 磯崎 善政

【発明者】

    【住所又は居所】 静岡県浜松市中沢町10番1号 ヤマハ株式会社内

    【氏名】 鈴木 秀雄

【発明者】

    【住所又は居所】 静岡県浜松市中沢町10番1号 ヤマハ株式会社内

    【氏名】 清水 正宏

【特許出願人】

    【識別番号】 000004075

    【住所又は居所】 静岡県浜松市中沢町10番1号

    【氏名又は名称】 ヤマハ株式会社

    【代表者】 上島 清介

【代理人】

    【識別番号】 100081880

    【弁理士】

    【氏名又は名称】 渡部 敏彦

    【電話番号】 03(3580)8464

【手数料の表示】

【予納台帳番号】 007065

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9202766

【書類名】 明細書

【発明の名称】 コンピュータソフトウェアを用いた音源システムおよび音響波形データ生成方法

【特許請求の範囲】

【請求項1】 楽音波形生成演算を行う、ソフトウェアで構成された楽音波形生成ブロックを複数個組み合わせることにより、複数の発音チャンネルの各チャンネル毎に楽音波形データを生成する音源システムにおいて、

前記組み合わせられる楽音波形生成ブロックの個数およびその組合せ態様を決定するアルゴリズムを設定するアルゴリズム設定手段と、

前記複数の発音チャンネルから楽音波形データを生成するチャンネルを選択する発音チャンネル選択手段と、

該選択された発音チャンネルに、前記設定されたアルゴリズムが示す楽音波形生成ブロックを割り当て、該割り当てられた楽音波形生成ブロックを当該アルゴリズムが示す組み合わせ態様で組み合わせて楽音波形生成演算を行い、楽音波形データを生成する楽音波形データ生成手段と  
を有することを特徴とする音源システム。

【請求項2】 楽音波形生成演算を行う、ソフトウェアで構成された楽音波形生成ブロックを複数個組み合わせることにより、複数の発音チャンネルの各チャンネル毎に異なった音色の楽音波形データを生成する音源システムにおいて、

前記組み合わせられる楽音波形生成ブロックの個数およびその組合せ態様を前記音色毎に決定するアルゴリズムを設定するアルゴリズム設定手段と、

前記複数の発音チャンネルから楽音波形データを生成するチャンネルを選択する発音チャンネル選択手段と、

該選択された発音チャンネルに、当該生成すべき楽音波形データの音色に対応して前記設定されたアルゴリズムが示す楽音波形生成ブロックを割り当て、該割り当てられた楽音波形生成ブロックを当該アルゴリズムが示す組み合わせ態様で組み合わせて楽音波形生成演算を行い、楽音波形データを生成する楽音波形データ生成手段と  
を有することを特徴とする音源システム。

【請求項3】 曲の各演奏パート毎に音色を設定する音色設定手段と、  
該各演奏パート毎に前記組み合わせられる楽音波形生成ブロックの個数を設定するブロック数設定手段と、

前記音色設定手段により音色が設定されるときに、当該演奏パートに前記ブロック数設定手段により楽音波形生成ブロック数が設定されている場合には、当該演奏パートに設定された音色を、該設定ブロック数の範囲内の楽音波形生成ブロックを備えた音色に変更する音色変更手段と  
を有することを特徴とする請求項2記載の音源システム。

【請求項4】 前記発音チャンネルにおける楽音波形生成演算中に、所定の条件に応じて、当該発音チャンネルに割り当てられた楽音波形生成ブロックの個数を変更するブロック数変更手段を有することを特徴とする請求項3に記載の音源システム。

【請求項5】 曲の各演奏パート毎に、該各パートにそれぞれ設定された音色に対応するアルゴリズムが示す楽音波形生成ブロックの削減態様を設定する削減態様設定手段を有し、

前記ブロック数変更手段は、該設定された削減態様および前記所定の条件に応じて楽音波形生成ブロックの個数を変更することを特徴とする請求項4記載の音源システム。

【請求項6】 前記所定の条件は、当該発音チャンネルにおいて生成された楽音波形の音量エンベロープが減衰したことであり、

前記ブロック数変更手段は、該楽音波形の音量エンベロープが所定値以下に減衰したときに、その減衰に応じて、前記アルゴリズム設定手段により設定されたアルゴリズムが示す楽音波形生成ブロックの個数を減少させる方向に変更することを特徴とする請求項4記載の音源システム。

【請求項7】 前記所定の条件は、当該発音チャンネルにおいて生成すべき楽音波形の音量が小さく設定されていることであり、

前記ブロック数変更手段は、当該楽音波形の音量が小さく設定されているときには、前記アルゴリズム設定手段により設定されたアルゴリズムが示す楽音波形生成ブロックの個数を減少させる方向に変更することを特徴とする請求項4記載

の音源システム。

【請求項 8】 前記所定の条件は、当該発音チャンネルに発音割り当てされた楽音波形生成ブロック中、その出力レベルが所定値より小さいブロックがあることであり、

前記ブロック数変更手段は、出力レベルが所定値以下となった楽音波形生成ブロックが他の楽音波形生成ブロックに影響を与えないものであるときには、当該楽音波形生成ブロックの演算を停止することにより、当該アルゴリズムが示す楽音波形生成ブロックの個数を減少させる方向に変更することを特徴とする請求項 4 記載の音源システム。

【請求項 9】 前記各楽音波形生成ブロック毎に、該各楽音波形生成ブロックが実行する楽音波形生成演算の基準となるサンプリング周波数を設定するサンプリング周波数設定手段を有し、

前記楽音波形データ生成手段は、前記各楽音波形生成ブロックによる楽音波形生成演算を、該各楽音波形生成ブロックに設定されたサンプリング周波数を基準にして行うことを特徴とする請求項 1 乃至 8 のいずれかに記載の音源システム。

【請求項 10】 所定の波形演算アルゴリズムを実行することにより、音響波形データを生成する音響波形データ生成方法において、

前記波形演算アルゴリズムを実行する演算部に関する所定の条件に応じて、前記音響波形データの生成に関するサンプリング周波数が変更されるようにしたことを特徴とする音響波形データ生成方法。

【請求項 11】 複数ステップから成る所定の波形演算アルゴリズムを実行しその各ステップにおける演算結果に基づいて、音響波形データを生成する音響波形データ生成方法において、

実行される前記波形演算アルゴリズム内での演算結果に応じてサンプリング周波数が変更されるようにしたことを特徴とする音響波形データ生成方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、ソフトウェアで記述された楽音波形生成ブロックを組み合わせ、該

各楽音波形生成ブロックによりそれぞれなされる楽音波形生成演算に基づいて楽音波形データを生成する音源システムおよび音響波形データ生成方法に関する。

【0002】

【従来の技術】

従来、たとえば波形メモリ音源方式やFM音源方式等の各種楽音生成方式に応じて楽音を生成するときには、当該楽音生成方式を実現する回路を専用のハードウェア（たとえば音源専用LSIや固定されたマイクロプログラムに基づいて動作するDSP（Digital Signal Processor）等）で構成していた。このように、専用のハードウェアで構成した楽音生成装置やその楽音生成方法を、以下、「ハード音源」という。

【0003】

しかし、ハード音源では、専用のハードウェアを必要とするため、製造コストをより低減させることは困難であり、また、設計後の仕様変更に対応することができないという問題があった。

【0004】

これに対して、近年、CPUの演算能力の向上に伴い、汎用コンピュータや専用の楽音発生装置に搭載されたCPUに対して、所定の楽音生成処理手順を記述したソフトウェアプログラムを実行させることにより、楽音波形データを生成させるようにしたものが開発されている。このように、ソフトウェアプログラムを中心に構成した楽音生成装置やその楽音生成方法を、以下、「ソフト音源」という。

【0005】

【発明が解決しようとする課題】

上記「従来の技術」においても述べたように、コンピュータシステム、コンピュータ応用機器でハード音源を用いるのは、コストの増大を招いたり、仕様変更に対する柔軟性が乏しくなるなどの問題があった。

【0006】

また、ソフト音源とはいっても、従来使われてきた音源LSIなどの専用ハードウェアやデバイスの機能をそのまま単純にソフトウェア化しただけのソフト音



源では、設計後の仕様変更によりハード音源より柔軟に対応することができるものの、発音中、すなわち音源動作中に発生する各種要請、たとえばCPUの能力やシステム環境からの要請およびユーザの好みや設定等の要請に柔軟に対応することができなかった。具体的には、出力された楽音波形の精度の変更（高精度への変更のみならず低精度への変更をも含む）やその音色変化の程度の変更（たとえば通常の音色変化より微妙な音色変化への変更やその逆方向への音色変化へ変更）等の要請に常に対応することができなかった。

#### 【0007】

本発明は、上記問題に鑑みてなされたもので、特別な専用ハードウェアを追加せずにソフトウェアプログラムで楽音生成を行うことによりコストを低減化させるとともに、その用途、動作環境、目的および設定等に応じて楽音生成の演算処理に対する演算手段の負荷の軽重および出力楽音波形の品質の高低を変更することが可能なコンピュータソフトウェアを用いた音源システムおよび音響波形データ生成方法を提供することを目的とする。

#### 【0008】

##### 【課題を解決するための手段】

上記目的を達成するため、本発明の音源システムは、楽音波形生成演算を行う、ソフトウェアで構成された楽音波形生成ブロックを複数個組み合わせることにより、複数の発音チャンネルの各チャンネル毎に楽音波形データを生成する音源システムにおいて、前記組み合わせられる楽音波形生成ブロックの個数およびその組合せ態様を決定するアルゴリズムを設定するアルゴリズム設定手段と、前記複数の発音チャンネルから楽音波形データを生成するチャンネルを選択する発音チャンネル選択手段と、該選択された発音チャンネルに、前記設定されたアルゴリズムが示す楽音波形生成ブロックを割り当て、該割り当てられた楽音波形生成ブロックを当該アルゴリズムが示す組み合わせ態様で組み合わせて楽音波形生成演算を行い、楽音波形データを生成する楽音波形データ生成手段とを有することを特徴とする。

#### 【0009】

また、楽音波形生成演算を行う、ソフトウェアで構成された楽音波形生成プロ

ックを複数個組み合わせることにより、複数の発音チャンネルの各チャンネル毎に異なった音色の楽音波形データを生成する音源システムにおいて、前記組み合わせられる楽音波形生成ブロックの個数およびその組合せ態様を前記音色毎に決定するアルゴリズムを設定するアルゴリズム設定手段と、前記複数の発音チャンネルから楽音波形データを生成するチャンネルを選択する発音チャンネル選択手段と、該選択された発音チャンネルに、当該生成すべき楽音波形データの音色に対応して前記設定されたアルゴリズムが示す楽音波形生成ブロックを割り当て、該割り当てられた楽音波形生成ブロックを当該アルゴリズムが示す組み合わせ態様で組み合わせて楽音波形生成演算を行い、楽音波形データを生成する楽音波形データ生成手段とを有することを特徴とする。

#### 【0010】

好ましくは、曲の各演奏パート毎に音色を設定する音色設定手段と、該各演奏パート毎に前記組み合わせられる楽音波形生成ブロックの個数を設定するブロック数設定手段と、前記音色設定手段により音色が設定されるときに、当該演奏パートに前記ブロック数設定手段により楽音波形生成ブロック数が設定されている場合には、当該演奏パートに設定された音色を、該設定ブロック数の範囲内の楽音波形生成ブロックを備えた音色に変更する音色変更手段とを有することを特徴とする。

#### 【0011】

また、好ましくは、前記発音チャンネルにおける楽音波形生成演算中に、所定の条件に応じて、当該発音チャンネルに割り当てられた楽音波形生成ブロックの個数を変更するブロック数変更手段を有することを特徴とする。

#### 【0012】

また、好ましくは、曲の各演奏パート毎に、該各パートにそれぞれ設定された音色に対応するアルゴリズムが示す楽音波形生成ブロックの削減態様を設定する削減態様設定手段を有し、前記ブロック数変更手段は、該設定された削減態様および前記所定の条件に応じて楽音波形生成ブロックの個数を変更することを特徴とする。

#### 【0013】

さらに、好ましくは、前記所定の条件は、当該発音チャンネルにおいて生成された楽音波形の音量エンベロープが減衰したことであり、前記ブロック数変更手段は、当該楽音波形の音量エンベロープが所定値以下に減衰したときに、その減衰に応じて、前記アルゴリズム設定手段により設定されたアルゴリズムが示す楽音波形生成ブロックの個数を減少させる方向に変更することを特徴とする。

## 【0014】

また、さらに好ましくは、前記所定の条件は、当該発音チャンネルにおいて生成すべき楽音波形の音量が小さく設定されていることであり、前記ブロック数変更手段は、当該楽音波形の音量が小さく設定されているときには、前記アルゴリズム設定手段により設定されたアルゴリズムが示す楽音波形生成ブロックの個数を減少させる方向に変更することを特徴とする。

## 【0015】

さらに、前記所定の条件は、当該発音チャンネルに発音割り当てされた楽音波形生成ブロック中、その出力レベルが所定値より小さいブロックがあることであり、前記ブロック数変更手段は、出力レベルが所定値以下となった楽音波形生成ブロックが他の楽音波形生成ブロックに影響を与えないものであるときには、当該楽音波形生成ブロックの演算を停止することにより、当該アルゴリズムが示す楽音波形生成ブロックの個数を減少させる方向に変更することを特徴とする。

## 【0016】

また、好ましくは、前記各楽音波形生成ブロック毎に、該各楽音波形生成ブロックが実行する楽音波形生成演算の基準となるサンプリング周波数を設定するサンプリング周波数設定手段を有し、前記楽音波形データ生成手段は、前記各楽音波形生成ブロックによる楽音波形生成演算を、該各楽音波形生成ブロックに設定されたサンプリング周波数を基準にして行うことを特徴とする。

## 【0017】

また、本発明の音響波形データ生成方法は、所定の波形演算アルゴリズムを実行することにより、音響波形データを生成する音響波形データ生成方法において、前記波形演算アルゴリズムを実行する演算部に関する所定の条件に応じて、前記音響波形データの生成に関するサンプリング周波数が変更されるようにしたこ

とを特徴とする。

【0018】

さらに、複数ステップから成る所定の波形演算アルゴリズムを実行しその各ステップにおける演算結果に基づいて、音響波形データを生成する音響波形データ生成方法において、実行される前記波形演算アルゴリズム内での演算結果に応じてサンプリング周波数が変更されるようにしたことを特徴とする。

【0019】

【発明の実施の形態】

以下、本発明の実施の形態を図面に基づいて詳細に説明する。

【0020】

図1は、本発明の実施の一形態に係る音源システムが備えたソフトウェアのシステム構成の一例を示す図であり、同図に示すように、本ソフトウェアシステムは、OSを介して、すなわちOS側で楽音波形データを生成するように構成されている。なお、同図には、OS側で生成されたデジタル楽音信号（楽音波形データ）をアナログ楽音信号に変換するDAC（Digital to Analog Converter）を含むc o d e cハードウェアも図示されている。

【0021】

図1において、AP S 1は、M I D Iメッセージ、すなわち楽音波形データを生成する基となる演奏情報をリアルタイムかつシーケンシャルに発生するアプリケーションソフトウェア（以下、「シーケンスソフト」という）である。該シーケンスソフトAP S 1は、曲毎に予め作成されたM I D Iファイル（たとえば各種イベントデータおよびそのイベントを発生させるタイミングデータ等のM I D Iデータから成るファイル）を複数個有し、その中からユーザにより1つまたは複数のM I D Iファイルが選択されると、該ファイルから順次M I D Iデータを読み出してM I D Iメッセージ（イベントデータに対応して生成されたM I D Iメッセージ）をリアルタイムかつシーケンシャルに発生する。そして、シーケンスソフトAP S 1は、該M I D Iメッセージを、M I D Iメッセージを入力するためにOS側が備えた第1のインターフェースI F 1（MIDI APplication Inter face; MIDI API）に出力する。

## 【0022】

OSには、第1のインターフェースIF1を介して入力されたMIDIメッセージに基づいて楽音波形データを生成するためのプログラム（ソフトウェア）であるソフトウェア音源モジュールSSMがドライバとしてインストールされるとともに、該ソフトウェア音源モジュールSSMにより生成された楽音波形データを入力するための第2のインターフェースIF2（WAVE out Application Interface; WAVE out API）も備わっている。さらに、OSには、第2のインターフェースIF2を介して入力された楽音波形データを外部に出力するための出力デバイスOUDもドライバとしてインストールされており、この出力デバイスOUDは、たとえば、ソフトウェア音源モジュールSSMの処理によって生成されてハードディスク等の記憶装置にバッファ記憶された楽音波形データをダイレクトメモリアクセス（DMA）コントローラを介して読み出し、所定のハードウェアやデバイス（たとえば前記c o d e cハードウェア）に対して出力する処理を行うソフトウェアにより構成される。

## 【0023】

シーケンスソフトAP S 1が出力したMIDIメッセージは、第1のインターフェースIF1およびOSを介してソフトウェア音源モジュールSSMの入力インターフェースに供給され、ソフトウェア音源モジュールSSMでは、受け取ったMIDIメッセージに基づく楽音波形データ生成処理（本実施の形態では、楽音波形データはFM音源方式により生成される）が実行され、この生成された楽音波形データは、第2のインターフェースIF2およびOSを介して出力デバイスOUDに供給される。出力デバイスOUDでは、供給された生成済みデジタル楽音波形データが前記c o d e cハードウェアに出力されて、アナログ楽音信号に変換される。

## 【0024】

このように、本実施の形態では、楽音波形データを生成するためのソフトウェア音源モジュールSSMとMIDIメッセージを出力するためのアプリケーションソフトウェアであるシーケンスソフトAP S 1とをOSレベルで簡単に組み合わせることができるようにしたので、楽音波形データを生成するための専用のハ

ードウェアを追加する必要がなく、これによりコストを低減させることができる。

【0025】

図2は、本実施の形態の音源システムを実現するハードウェアの概略構成を示す図である。本システムは、汎用のパーソナルコンピュータにより実現され、そのメイン制御部としてはCPU3が使用され、該CPU3の制御の下で、ソフトウェア音源プログラムによる楽音波形データ生成処理と該楽音波形データ生成処理以外のその他のプログラム処理とが並行して実行される。

【0026】

同図において、CPU3には、外部からのMIDIメッセージを入力したり、MIDIメッセージを外部に出力したりするMIDIインターフェースMIDI I/F1、タイマ割込時間や各種時間を計時するタイマ2、各種制御プログラムやテーブルデータ等を記憶するROM（リードオンリメモリ）4、前記選択されたMIDIファイルや各種入力情報および演算結果等を一時的に記憶するRAM（ランダムアクセスメモリ）5、ポインティングデバイスであるマウス7、主として文字情報を入力するためのキーボード8、たとえば大型LCDやCRT等の各種情報等を表示するディスプレイ9、CPU3が実行する各種制御プログラムを含む各種アプリケーションプログラムや各種データ等を記憶するハードディスクをドライブするハードディスク装置10、DMA（Direct Memory Access）コントローラ11、および、通信ネットワーク101を介して、たとえばサーバコンピュータ102とデータの送受信を行う通信インターフェース（I/F）14が、データおよびアドレスバス6を介して接続されている。

【0027】

DMAコントローラ11は、楽音生成処理の実行により生成され、RAM5内の出力バッファに書き込まれた楽音データを、DAC12に内包されるデータバッファ（図示せず）の空き状態などに応じてダイレクトメモリアクセス方式で出力バッファから直接読み出してDAC12の内包データバッファに転送する処理（再生処理）を実行するものである。DAC12でアナログ変換された楽音データは、サウンドシステム13に送られ、該システム13により音響に変換される。

【0028】

ハードディスク装置10内のハードディスクには、前記OSやユーティリティソフト等のソフトウェアの他に、ソフトウェア音源を実現するためのソフトウェア（すなわち前記ソフトウェア音源モジュールSSM）と、その他のアプリケーションソフト（すなわち前記シーケンスソフトAPS1）が記憶されている。

【0029】

前記出力デバイスOUDは、ソフトウェア音源モジュールSSMからOSレベルの前記第2のインターフェースIF2を介して供給された楽音データをDAC12に送るモジュールに相当する。前述のように、DMAコントローラ11がダイレクトメモリアクセス方式で楽音データをDAC12に送るようになっているので、出力デバイスOUDは、CPU3の制御の下で、DMAコントローラ11による割込み処理として実行される。

【0030】

通信I/F14は、LAN（ローカルエリアネットワーク）やインターネット、電話回線等の通信ネットワーク101に接続されており、該通信ネットワーク101を介して、サーバコンピュータ102と接続される。ハードディスク装置10内のハードディスクに上記各プログラムや各種パラメータが記憶されていない場合には、通信I/F14は、サーバコンピュータ102からプログラムやパラメータをダウンロードするために用いられる。クライアントとなるコンピュータ（本実施の形態の音源システム）は、通信I/F14及び通信ネットワーク101を介してサーバコンピュータ102へとプログラムやパラメータのダウンロードを要求するコマンドを送信する。サーバコンピュータ102は、このコマンドを受け、要求されたプログラムやパラメータを、通信ネットワーク101を介してコンピュータへと配信し、コンピュータが通信I/F101を介して、これらプログラムやパラメータを受信してハードディスク装置10内のハードディスクに蓄積することにより、ダウンロードが完了する。

【0031】

この他、外部コンピュータ等との間で直接データのやりとりを行うためのイン

ターフェースを備えてもよい。

【0032】

次に、ソフトウェア音源モジュールSSMによるFM音源方式を用いた楽音生成処理の概要を、図3～6を参照して説明する。

【0033】

シーケンスソフトAP S 1が起動されると、ソフトウェア音源モジュールSSMにMIDIメッセージが供給され始める。この供給は、前述したように、第1のインターフェースIF 1およびOSを介して、ソフトウェア音源モジュールSSMの入力インターフェース（以下、「ソフト音源インターフェース」という）に対してなされ、これに応じて、ソフトウェア音源モジュールSSMは、その供給されたMIDIメッセージのMIDIチャンネル（演奏パート）に割り当てられたボイスデータ（本実施の形態では、音色番号）に基づいて、楽音パラメータVOICE j（その詳細は後述する）を生成し、該楽音パラメータVOICE jを発音チャンネル（発音割り当てされたチャンネル）に対応する音色レジスタに格納する。

【0034】

図6は、発音チャンネルに対応して設けられた音色レジスタ群を示す図であり、発音チャンネルが、たとえば32チャンネル確保されているときには、この音色レジスタ群は32個の音色レジスタTONE P A R k（ $k=1, \dots, 32$ ）により構成される。なお、発音チャンネル数は、これに限る必要はなく、CPU 3の演算処理能力に応じて任意に設定することができる。

【0035】

同図において、たとえば発音チャンネルがチャンネルnである場合には、前記楽音パラメータVOICE jは、音色レジスタTONE P A R n中、該楽音パラメータVOICE jを格納する領域に格納される。換言すれば、これらの音色レジスタTONE P A R kにより構成される音色レジスタ群がソフトウェア音源モジュールSSMの前記ソフト音源インターフェースの一部をなしている。

【0036】

なお、後述するように、各音色レジスタTONE P A R kには、楽音パラメー



タVOICE<sub>k</sub>だけではなく、たとえば、ソフトウェア音源モジュールSSMが当該楽音パラメータVOICE<sub>k</sub>に対応するMIDIメッセージを受信した時刻を示すデータTMも格納され、フレーム内での楽音のキーオン、キーオフあるいは各種制御の開始、終了の時間位置を決めるための情報となる。

#### 【0037】

図3に戻り、ソフトウェア音源モジュールSSMは、CPU3の制御の下で、基本的には、所定の時間長の区間（以下、「フレーム」という）毎に設定される起動機会に応じて起動され、その直前のフレーム内に供給されたMIDIメッセージに基づく楽音生成処理を、上記音色レジスタTONEPAR<sub>n</sub>に格納された楽音パラメータVOICE<sub>n</sub>に従って実行する。たとえば、図3において、時刻 $t_1$ から $t_2$ までのフレーム内に供給されたMIDIメッセージに基づく楽音生成処理は、時刻 $t_2$ から $t_3$ までのフレームで実行する。

#### 【0038】

このようにして、楽音生成処理（その処理方法については、後述する）により1フレーム分の楽音波形データが生成されると、この楽音波形データはRAM5の前記出力バッファに書き込まれ、該書き込まれたデータの再生が前記出力デバイスOUDに予約される。この出力デバイスOUDに予約することが、生成した楽音波形データをソフトウェア音源モジュールSSMからOSレベルの前記第2のインターフェース（“WAVE out API”）に出力することに相当する。

#### 【0039】

出力デバイスOUDは、各フレーム毎に、その直前のフレームで再生予約された出力バッファから楽音波形データを1サンプルずつ読み出して前記DAC12に出力する。たとえば、図3に示すように、時刻 $t_2$ から $t_3$ までのフレームで生成され、出力バッファに書き込まれて再生予約された楽音波形データは、時刻 $t_3$ から $t_4$ までのフレームで読み出されて再生される。

#### 【0040】

次に、図4を用いて、楽音パラメータVOICE<sub>n</sub>に従った楽音生成処理、すなわちFM音源方式を用いた楽音生成処理の概要を説明する。なお、この楽音生成処理の詳細は、図14～19を用いて後述する。

【0041】

図4は、FM音源方式による楽音生成処理の概要を説明するための図であり、同図(a)～(c)には、3種類の異なった楽音生成処理方法が図示されている。

【0042】

図4に示すように、FM音源方式による楽音生成は、「キャリア」と呼ばれるオペレータと「モジュレータ」と呼ばれるオペレータの2種類のオペレータを組み合わせることによって行われ、生成すべき楽音波形の種類や品質等に応じて組み合わせられるオペレータの個数や接続順序（接続態様）が異なる。この各オペレータのつながぎ方を「アルゴリズム」と呼んでいる。

【0043】

ここで、オペレータとは、音作り、すなわち楽音生成処理の単位となるブロックをいい、具体的には、たとえば図5に示す音作りの基本となる各種波形データ（以下、「基本波形データ」という）から、後述するウェーブセレクトパラメータWSELに応じて選択された1つの基本波形データを、入力データ（たとえばピッチデータおよび変調データ）に基づいて（入力データがピッチデータおよび変調データの2種類のデータである場合には、たとえば該各データを加算した加算結果に基づいて）読み出した後にその振幅を調整して出力する処理ブロックをいう。そして、「キャリア」とは、生成すべき楽音波形の基準となる楽音波形を生成するオペレータを云い、「モジュレータ」とは、「キャリア」を変調するためのオペレータ、すなわち「キャリア」に入力される前記変調データを生成するオペレータを云う。

【0044】

なお、アルゴリズムは、図4に示す3種類に限られないことは云うまでもない。

【0045】

次に、前記楽音パラメータVOICEjのデータフォーマットについて説明する。

【0046】

図7は、この楽音パラメータVOICE<sub>j</sub>のデータフォーマットを示す図であり、(a)は、楽音パラメータVOICE<sub>j</sub>のデータフォーマットを示し、(b)は、(a)の各オペレータデータOP<sub>m</sub>DATA<sub>j</sub>のデータフォーマットを示し、(c)は、(b)の各オペレータバッファOPBUF<sub>m</sub>のデータフォーマットを示している。

【0047】

(a)に示すように、楽音パラメータVOICE<sub>j</sub>は、キーオン／オフをそれぞれ“1”／“0”で示すキーオンデータKEYON<sub>j</sub>と、当該MIDIメッセージ（すなわち、対応するノートオンイベント）に含まれる音高情報に応じて決定される周波数ナンバ（具体的には、位相レートで表される）FNO<sub>j</sub>と、前記アルゴリズムを指定するアルゴリズム指定データALGOR<sub>j</sub>と、当該MIDIチャンネルに設定された音量（たとえば、MIDIメッセージの「コントロールチェンジ#7イベント」により設定される）に応じて決定される音量データVOL<sub>j</sub>と、当該MIDIメッセージ中のタッチベロシティ情報に応じて決定されるタッチベロシティデータVEL<sub>j</sub>と、各構成オペレータにおいて楽音生成演算を行うために必要なデータおよびその演算結果を格納するバッファから成るオペレータデータOP<sub>k</sub>DATA<sub>j</sub>（ $k=1, \dots, m$ ）とにより構成されている。

【0048】

なお、(a)の楽音パラメータVOICE<sub>j</sub>には、前記ROM4、RAM5またはハードディスク等に予め記憶されたデータを読み出したデータとMIDIメッセージ中のデータに応じて決定されるデータの2種類のデータが混在している。MIDIメッセージ中のデータに応じて決定されるデータは、上述のように、キーオンデータKEYON<sub>j</sub>、周波数ナンバFNO<sub>j</sub>、音量データVOL<sub>j</sub>およびタッチベロシティデータVEL<sub>j</sub>であり、ROM4等から読み出されたデータは、アルゴリズム指定データALGOR<sub>j</sub>およびオペレータデータOP<sub>k</sub>DATA<sub>j</sub>である。

【0049】

各オペレータデータOP<sub>k</sub>DATA<sub>j</sub>は、(b)に示すように、オペレータ<sub>m</sub>におけるサンプリング周波数を指定するサンプリング周波数指定データFSAM

P<sub>m</sub>と、音色に対応して設定され、実質的に各オペレータ間の周波数比を設定するためのパラメータ、具体的には前記周波数ナンバFNO<sub>j</sub>を変倍する倍数を指定するためのパラメータである周波数倍数データMULT<sub>m</sub>と、フィードバックレベル、すなわちフィードバック変調の変調度を示すフィードバックレベルデータFBL<sub>m</sub>と、ROM4等に予め記憶された各種基本波形データ（前記図5で説明した各種基本波形データ）からオペレータ<sub>m</sub>で使用する基本波形データを選択するためのウェーブセレクトデータWSEL<sub>m</sub>と、オペレータ<sub>m</sub>において生成される楽音波形の出力レベル（このレベルは、前記タッチベロシティデータVEL<sub>j</sub>に応じて変化する）を設定するトータルレベルデータTL<sub>m</sub>と、オペレータ<sub>m</sub>において生成される楽音波形のエンベロープを決定する各種データ（たとえば、アタックタイム、ディケイタイム、サステインレベルおよびリリースタイム等の各種データ）から成るエンベロープパラメータEGPAR<sub>m</sub>と、その他パラメータ（たとえば、ビブラートやトレモロ等の速さおよび深さ、各種キースケーリング係数等）MSC<sub>m</sub>と、オペレータ<sub>m</sub>の優先度（たとえば、各オペレータにおける波形生成演算やその停止の優先度）を示すオペレータ優先度データOPPRIO<sub>m</sub>と、オペレータ<sub>m</sub>における楽音波形生成演算による演算結果を格納するバッファOPBUF<sub>m</sub>とにより構成される。

#### 【0050】

ここで、サンプリング周波数指定データFSAMP<sub>m</sub>には、“0”以上の整数値 $f$ が格納され、標準モードにおけるサンプリング周波数FSMAX（たとえば44.1kHz）を $2^{-f}$ 倍に変更することができる。たとえば、 $f=0$ のときには、標準モードのサンプリング周波数FSMAXでオペレータ<sub>m</sub>における楽音波形が生成され、 $f=1$ のときには、FSMAX/2のサンプリング周波数でオペレータ<sub>m</sub>における楽音波形が生成される。

#### 【0051】

また、オペレータ優先度データOPPRIO<sub>m</sub>には、楽音パラメータVOICE<sub>j</sub>を構成する全オペレータ $k$ （ $k=1, \dots, m$ ）におけるオペレータ<sub>m</sub>の波形演算処理の優先度を示すデータ（たとえば波形演算処理を行う順序を示す番号）が格納され、この優先度に応じて各オペレータの演算順序が決定され、波形演算

処理がなされる。しかし、これに限らず、たとえばCPU3の能力や負荷状態を確認できるようにし、この結果、CPU3が音源処理をする余裕がない場合には、優先度の低いオペレータの演算処理を省略するようにしてもよい。さらに、本実施の形態では、演算処理の優先度は、音色に応じて設定するようにしたが、これに限らず、たとえばMIDIチャンネルに応じて設定するようにしてもよい。要するに、優先度としては、何らかの基準に従って設定したものを選択して設定し、該優先度を発音時に使用するようにすればよい。たとえば、優先度が音色に応じて設定されていない場合には、前記音色レジスタTONEPAR<sub>n</sub>に展開された音色パラメータの各オペレータ優先度データOPPRIO<sub>m</sub>を、その優先度に応じて決定するようにすればよい。また、オペレータ優先度データOPPRIO<sub>m</sub>は、オペレータ<sub>m</sub>を使用する／しないの設定として扱うことにしてもよい。

#### 【0052】

なお、本実施の形態では、サンプリング周波数は、上記サンプリング周波数指定データFSAMP<sub>m</sub>により、各オペレータ<sub>m</sub>毎に設定することができるように構成したが、これに限らず、「キャリア」および「モジュレータ」という2種類の範疇毎にそれぞれ設定するようにしてもよい。たとえば、「キャリア」は、前記周波数FSMAXとし、「モジュレータ」は、その1/2倍(=FSMAX/2)の周波数とした場合には、当該音色パラメータのアルゴリズムの内容をチェックし、組み合わせられるオペレータに対して、それぞれのサンプリング周波数を設定すればよい。また、CPU3の負荷状態をチェックし、その負荷状態に応じて、適宜サンプリング周波数を増大または低下させるようにしてもよい。

#### 【0053】

前記バッファOPBUF<sub>m</sub>は、(c)に示すように、オペレータ<sub>m</sub>により波形演算を行うこと、すなわちオペレータ<sub>m</sub>がオンであることを“1”で示すオペレータオンパラメータOPON<sub>m</sub>と、オペレータ<sub>m</sub>が実行する波形演算処理(図14および15を用いて後述する)の位相演算処理により算出された位相値を格納する位相値バッファPHBUF<sub>m</sub>と、該波形演算処理のフィードバックサンプル演算処理により算出されるフィードバック出力値を格納するフィードバック出力

値バッファFB<sub>m</sub>と、前記変調データ（このデータは上記位相演算処理において使用する）を格納する変調データ入力バッファMODIN<sub>m</sub>と、オペレータ<sub>m</sub>により生成された楽音波形、すなわち出力値を格納するオペレータ出力値バッファOPOUT<sub>m</sub>と、上記波形演算処理の振幅制御用EGを算出する演算処理（以下、「AEG演算処理」という）により算出された各EGパラメータを格納するEG状態バッファEGSTATE<sub>m</sub>とにより構成されている。

## 【0054】

図8は、各MIDIチャンネル毎、すなわち各パート毎に選択設定されたボイス（音色）データ（本実施の形態では、楽音パラメータVOICE<sub>n</sub>の音色番号）を記憶するMIDI-CHボイステーブルを示す図である。

## 【0055】

同図に示すように、本実施の形態では、MIDIチャンネルは16個のチャンネルで構成され、各チャンネル、すなわち各パートにはそれぞれ異なった音色を設定することができるので、本実施の形態の音源システムは、最大16種類の音色の楽音を生成することができる。そして、このMIDI-CHボイステーブルには、各チャンネル毎に選択された音色に対応する音色番号、すなわち前記楽音パラメータVOICE<sub>n</sub>に付与された番号が格納される。

## 【0056】

なお、MIDI-CHボイステーブルは、前記RAM5の所定領域に確保され、そのテーブルデータ、すなわち音色番号は、前記各MIDIファイルに対応して予めハードディスク等に格納され、ユーザが選択したMIDIファイルが、RAM5の所定位置に確保された演奏データ格納領域にロードされると同時に、該MIDIファイルに対応するテーブルデータもMIDI-CHボイステーブルにロードされる。もちろんこれに限らず、ユーザが最初から任意に設定できるようにしてもよいし、曲に標準の音色（番号）が予め設定され、このテーブルデータがMIDI-CHボイステーブルにロードされた後に、ユーザが任意に変更することができるようにしてもよい。

## 【0057】

前述のように、MIDIメッセージが1つずつシーケンシャルに生成され、ソ

ソフトウェア音源モジュールSSMにより、このMIDIメッセージが認識されると、ソフトウェア音源モジュールSSMは、該MIDIメッセージのMIDIチャンネルに割り当てられた音色番号を前記MIDI-CHボイステーブルから検索する。たとえば、該MIDIメッセージのMIDIチャンネルが「2CH」のときには、MIDI-CHボイステーブルの2番目の位置VOICENO2に格納された音色番号が検索される。

## 【0058】

音色番号jが検索されると、ソフトウェア音源モジュールSSMは、前述したように、楽音パラメータVOICEjを生成する。すなわち、基本データをROM4等から読み出すとともに、その他のパラメータを当該MIDIメッセージから決定して、前記図7に示す楽音パラメータVOICEjを作成する。そして、ソフトウェア音源モジュールSSMは、このようにして作成された楽音パラメータVOICEjを、図6に示す音色レジスタ群中、発音割り当てされた発音チャンネルに対応する音色レジスタ（たとえば音色レジスタTONEPARn）に展開する。

## 【0059】

以上のように構成された音源システムが実行する制御処理を、以下、図9～20を参照して説明する。

## 【0060】

図9は、本実施の形態の音源システム、特にCPU3が実行する初期プログラム、すなわちユーザが音源システムの電源を投入したときやリセットスイッチを押下したときに実行されるプログラムの手順を示すフローチャートである。

## 【0061】

同図において、まず、各種ポートのリセットやRAM5およびディスプレイ9内の図示しないVRAM（ビデオRAM）のクリア等のシステムの初期化を行う（ステップS1）。

## 【0062】

次に、前記OSのプログラムを、たとえば前記ハードディスク装置10のハードディスクから読み出してRAM5の所定領域にロードし、該OSプログラムを

起動した（ステップS2）後に、次のメインプログラムへ移行する。

【0063】

図10は、上記初期プログラムに続いてCPU3が実行するメインプログラムの手順を示すフローチャートであり、本メインプログラムは、ソフトウェア音源モジュールSSMのプログラムのメインルーチンである。

【0064】

同図において、まず、RAM5中、ソフトウェア音源モジュールSSMで使用される領域（図6の音色レジスタ群を含む）のクリアや、ハードディスク装置10のハードディスクに記憶されている各種基本波形データ（たとえば図5の各種基本波形データ）をRAM5の所定領域にロードする等の初期設定を行う（ステップS11）。

【0065】

次に、ディスプレイ9に、処理の進行に応じた情報を表示したり、主としてマウス7により選択される各種アイコン等を表示したりする基本ディスプレイ表示を実行する（ステップS12）。

【0066】

そして、下記の各起動要因の発生をチェックする（ステップS13）。

【0067】

起動要因1：前記シーケンスソフトAPS1が起動され、MIDIメッセージがソフトウェア音源モジュールSSMに供給されたこと。

【0068】

起動要因2：ソフトタイマにより、波形演算処理（ソフトウェア音源モジュールSSMによりなされる）の実行を起動する内部割込信号（起動信号）が発生したこと。

【0069】

起動要因3：前記codecハードウェアから、前記出力バッファに格納された楽音波形データをcodecハードウェア内の図示しないバッファに転送する要求があったこと。

【0070】



起動要因4：ユーザがマウス7やキーボード8等の入力操作子を操作し、その操作イベントが検出されたこと、または他の起動要因が発生していないこと。

【0071】

起動要因5：ユーザがメインルーチンを終了する操作処理を行い、この操作イベントが検出されたこと。

【0072】

続くステップS14では、上記起動要因1～5のうちいずれかが発生したか否かを判別する。

【0073】

ステップS14の判別の結果、「起動要因1」が発生したときにはステップS16に進み、図11および12を用いて後述するMIDI処理サブルーチンを実行し、「起動要因2」が発生したときにはステップS17に進み、図14および15を用いて後述する波形演算処理サブルーチンを実行し、「起動要因3」が発生したときにはステップS18に進み、前記出力バッファに格納された楽音波形データを前記c o d e cハードウェアのバッファに転送し、「起動要因4」が発生したときにはステップS19に進み、音色設定イベントが発生した場合には図19を用いて後述する音色設定処理サブルーチンを実行するとともに、その他イベントが発生した場合には当該イベントに対応するその他処理を実行し、「起動要因5」が発生したときにはステップS21に進み、たとえばディスプレイ9の表示を本メインプログラムが起動される前の状態に戻す等の終了処理を実行する。

【0074】

そして、上記ステップS16～S21のいずれかを終了した後は、前記ステップS12に戻り前述の処理を繰り返す。

【0075】

図11および12は、前記ステップS16のMIDI処理サブルーチンの詳細な手順を示すフローチャートである。

【0076】

同図において、まず、ソフトウェア音源モジュールSSMの前記ソフト音源イ

インターフェースAPIを介してMIDIイベント（MIDIメッセージ）が入力されたか否かをチェックする（ステップS31）。

【0077】

シーケンスソフトAP S1からMIDIメッセージが出力されると、該MIDIメッセージは、前記第1のインターフェースIF1およびOSにより所定の変換がなされた後に、ソフト音源APIを介してRAM5の所定位置に確保されたMIDIイベントバッファに転送される。この転送に応じて、ソフトウェア音源モジュールSSMは、「起動要因1」が発生したと判断して、CPU3の制御をステップS15からS16に移行させる。ここまでの処理は、前記ステップS20のその他処理で行われるので、ステップS31では、このMIDIイベントバッファに格納されているイベントをチェックすることにより、イベントの発生をチェックしている。

【0078】

次に、ステップS32では、MIDIイベントがノートオンイベントであるかを判別し、ノートオンイベントのときにはステップS33に進む一方、ノートオンイベントでないときには図12のステップS40に進む。

【0079】

ステップS33では、当該ノートオンイベントデータを解析し、ノートナンバ、ベロシティ値およびパートナンバ（すなわちMIDIチャンネルの番号）の各データを、それぞれRAM5の所定位置に確保されたレジスタNN、VEL、pに格納するとともに、当該ノートオンイベントが発生した時刻を示すデータをRAM5の所定位置に確保されたレジスタTMに格納する。以下、レジスタNN、VEL、p、TMの内容を、それぞれノートナンバNN、ベロシティVEL、パートpおよび時刻TMという。

【0080】

続くステップS34では、ベロシティVELが所定値VEL1以下であり、かつ（&）音量データVOLpが所定値VOL1以下であるか否かを判別する。ここで、VOLpは、RAM5の所定位置に確保された領域VOLpに格納されたパートpの音量データを示し、この値は、前記図7で説明したように、MIDI

イベントのコントロールチェンジ#7イベントにより変更される。この変更処理は、コントロールチェンジ#7イベントが発生したときに、前記ステップS20のその他処理で行われる。

【0081】

ステップS34で、 $VEL \leq VEL1$  &  $VOL \leq VOL1$  のときには、パートpの音色に近い音色で、オペレータ数、すなわち前記キャリアおよびモジュレータの総数の小さいアルゴリズムの音色に置き換える、すなわち前記MIDI-CHボイステーブルのパートpの位置VOICENOPに格納された音色番号をその代替アルゴリズムを有する楽音パラメータVOICEの音色番号に置き換える（ステップS35）一方、 $VEL > VEL1$  または  $VOL > VOL1$  のときには、ステップS35をスキップしてステップS36に進む。

【0082】

なお、本実施の形態では、ステップS35の処理を行うか否かを、ベロシティVELおよびボリュームVOLの各値に応じて判別したが、これに限らず、たとえば、CPU3の負荷状態を検出し、この検出結果に応じて判別するようにしてもよい。

【0083】

ステップS36では、当該ノートオンイベントに基づく発音割り当て処理を行い、該割り当てられた発音チャンネルのチャンネル番号を、RAM5の所定位置に確保されたレジスタnに格納する。以下、レジスタnに格納された内容を発音チャンネルnという。

【0084】

続くステップS37では、前記図8のMIDI-CHボイステーブルを検索し、パートpの位置VOICENOPの音色データ（音色番号）を、前記ノートナンバNNおよびベロシティVELに応じて発音音色パラメータ（楽音パラメータ）に変換する。たとえば、位置VOICENOPに音色番号jが格納されている場合には、前記図7で説明した楽音パラメータVOICEjが生成される。そして、楽音パラメータVOICEjの各オペレータデータOPmDATAj内のバッファOPBUFmを初期化、すなわちクリアする。

## 【0085】

次に、ステップS38では、ステップS37で生成した楽音パラメータVOICE<sub>j</sub>を時刻TMとともに発音チャンネルnに対応する前記音色レジスタTONEPAR<sub>n</sub>に転送（展開）し、かつ該音色レジスタTONEPAR<sub>n</sub>中のキーオンデータKEYON<sub>n</sub>および各オペレータオンパラメータOPON<sub>m</sub>をそれぞれ“1「オン」”を設定にする。

## 【0086】

さらに、ステップS39では、ノートオンイベントの発生時刻が遅い順序に楽音生成演算が行われるように、発音割り当てされた各発音チャンネル間で演算順序を決定する。具体的には、この決定された演算順序に従ってチャンネル番号を並べ替え、RAM5の所定位置に確保された図13に示すCHシーケンスレジスタCHSEQに記憶する。その後、本MIDI処理を終了する。

## 【0087】

図12のステップS40では、MIDIイベントがノートオフイベントであるか否かを判別し、ノートオフイベントのときにはステップS41に進む一方、ノートオフイベントでないときにはステップS44に進む。

## 【0088】

ステップS41では、当該ノートオフイベントデータを解析し、そのノートナンバを前記レジスタNNに格納するとともに、当該ノートオフイベントが発生した時刻を示すデータを前記レジスタTMに格納する。

## 【0089】

続くステップS42では、ノートナンバNNが発音割り当てされている発音チャンネルをサーチし、そのチャンネル番号を、RAM5の所定位置に確保されたレジスタi（以下、この値を「発音チャンネルi」という）に格納する。

## 【0090】

そして、ステップS43では、発音チャンネルiに対応する音色レジスタTONEPAR<sub>i</sub>にキーオフを指定する、すなわち時刻TMに応じたタイミングでのノートオフの予約を行った後に、本MIDI処理を終了する。

## 【0091】

ステップS44では、MIDIイベントがプログラムチェンジイベント（音色を変更するためのイベント）であるか否かを判別し、プログラムチェンジイベントのときには、前記MIDI-CHボイステーブル中、当該受信したプログラムチェンジイベントが指定するパートp（このパートpは、前記ステップS33で格納されたパート番号であるとは限らない）に対応する位置VOICENOPのデータを、当該受信したプログラムチェンジイベントが指定する変更値PCHNGに変更した後に本MIDI処理を終了する一方、MIDIイベントがプログラムチェンジイベントでないときには、当該MIDIイベントに対応するその他イベント対応処理を実行した後に本MIDI処理を終了する。

#### 【0092】

なお、本MIDI処理では、MIDI-CHボイステーブルに複数パートに対応する音色が指定され、当該指定されたパートのノートオンイベントが発生したときには、当該パートの音色の楽音を生成して発音する、いわゆるマルチ音色動作仕様を採用しているが、これに限らず、ある特定のパートのノートオンイベントのみを受け付け、その対応の音色の楽音を生成して発音する、いわゆるシングル音色モードを選択できるようにしてもよい。

#### 【0093】

図14および15は、前記図10のステップS17の波形演算処理サブルーチンの詳細な手順を示すフローチャートである。

#### 【0094】

同図において、まず、楽音波形バッファの初期化を行う（ステップS51）。ここで、楽音波形バッファとは、前記出力バッファ領域から前記再生予約された領域（バッファ）を除いた領域中の、今回生成する1フレーム時間分の領域（バッファ）をいう。そして、その楽音波形バッファの初期化とは、当該領域を出力バッファ上に確保し、その領域をクリアすることをいう。

#### 【0095】

次に、CPU3の負荷状態をチェックし（ステップS52）、波形演算処理を実行可能な最大チャンネル数CHmaxを決定する（ステップS53）。ここで、CPU3の負荷状態のチェックは、OSがCPU3の負荷状態を常にチェック

している場合にはこの情報を利用して行えばよいし、一方、OSがCPU3の負荷状態をチェックしていない場合には、前記図10のメインプログラムを1回回る時間を計測するルーチンを設け、その計測値に応じて算出された値を利用して行えばよい。

【0096】

なお、ステップS53の処理に代えて、たとえば、前記図11のステップS35と同様の処理、すなわちパートに割り当てられた音色を、構成オペレータ数がより小さな（CPU3の能力によっては大きい場合もある）代替音色に変更する処理を行うようにしてもよい。

【0097】

次に、チャンネル番号を示す指標（インデックス） $i$ を初期化（ $i \leftarrow 1$ ）する（ステップS54）。

【0098】

そして、ステップS55では、前記図13のCHシーケンスレジスタCHSEQ中、 $i$ 番目の位置SEQCH $i$ に格納されたチャンネル番号SEQCHNO $i$ を変数 $n$ （本波形演算処理サブルーチン内において、この値を「チャンネル $n$ 」という）に格納し（ $n \leftarrow \text{SEQCHNO } i$ ）、ステップS56では、チャンネル $n$ に対応する音色レジスタTONEPAR $n$ のアルゴリズム指定データALGOR $n$ を参照し、チャンネル $n$ についてのFM演算処理（その詳細は図16を用いて後述する）で使用するオペレータ（OP）数と各オペレータの接続態様を決定する。

【0099】

さらに、ノートイベント等に応じて今回のフレームでの演算量を決定する（ステップS57）。ここで、演算量の決定とは、具体的には、チャンネル $n$ についての波形演算処理を、前記楽音波形バッファ中、どこからどこまでの領域を用いて行うかを決定することをいう。すなわち、楽音波形バッファは、前述したように、今回演算する1フレーム分の領域であり、一方、各チャンネルの楽音波形データは、1フレーム分の全領域に亘って生成される訳ではない、すなわち各チャンネル毎に楽音の発音タイミングや消音タイミングが異なり、あるチャンネルの

楽音は楽音波形バッファの途中で発音開始したり、消音されたりすることがあるため、各チャンネル毎に演算量を決定する意味がある。

【0100】

次いで、図15のステップS58では、図16を用いて後述するチャンネルnについて1サンプルの楽音波形データを生成するFM演算処理サブルーチンを実行し、ステップS59では、チャンネルnについて1フレーム分の楽音生成処理を完了したか否かを判別する。なお、ステップS59の判別は、前記ステップS57で決定された演算量を考慮して行われることは云うまでもない。

【0101】

ステップS59で、チャンネルnについて1フレーム分の楽音生成処理を完了していないときには、前記ステップS58に戻り、次の1サンプルの楽音波形データを生成する。一方、ステップS59で、チャンネルnについて1フレーム分の楽音生成処理を完了したときにはステップS60に進む。

【0102】

ステップS60では、ステップS58およびS59で演算され、生成された1フレーム分の楽音波形データを前記楽音波形バッファに書き込む。このとき、楽音波形バッファにすでに楽音波形データが格納されている場合には、演算されたデータを当該データに加算し、その加算結果を書き込む。

【0103】

そして、指標iの値を“1”だけインクリメントして更新し（ステップS61）、指標iの値が前記最大チャンネル数CHmaxより大きいかな否かを判別する（ステップS62）。

【0104】

ステップS62で、 $i \leq CHmax$ のとき、すなわち波形生成すべきチャンネルに対する処理が残っているときには、前記図14のステップS55に戻って前述の処理を繰り返す一方、 $i > CHmax$ のとき、すなわち波形生成すべきチャンネルに対する処理が終了したときには、今回ノートオフのあった発音チャンネルについての音量エンベロープの大きさを徐々に小さくする消音チャンネル処理を実行する（ステップS63）。

## 【0105】

続くステップS64では、このようにして生成された楽音波形データを、楽音波形バッファから切り離して、出力デバイスであるc o d e cハードウェアに渡し、再生指示（再生予約）した後に、本波形演算処理を終了する。

## 【0106】

なお、チャンネルnのベロシティ値が所定値より小さくなったときには、当該チャンネルnについてのFM演算を行わないようにしてもよい。これを実現するためには、図14に示すように、前記ステップS55の後にステップS71を設け、チャンネルnの音色レジスタTONEPAR<sub>n</sub>中のタッチベロシティデータVEL<sub>n</sub>が所定値VEL<sub>n1</sub>以上か否かを判別し、VEL<sub>n</sub> ≥ VEL<sub>n1</sub>のときには前記ステップS56に進む一方、VEL<sub>n</sub> < VEL<sub>n1</sub>のときには、ステップS72で、前記図12のステップS43と同様にしてチャンネルnのキーオフ指定をした後に前記ステップS61に進むようにすればよい。

## 【0107】

図16は、前記ステップS57のチャンネルnのについてのFM演算処理サブルーチンの詳細な手順を示すフローチャートである。

## 【0108】

同図において、まず、演算対象オペレータのオペレータ番号を格納するための変数m（以下、この値を「演算対象オペレータm」という）を初期化（m ← 1）する。

## 【0109】

次に、前記ステップS51と同様にして、CPU3の負荷状態をチェックするとともに、演算対象オペレータmのオペレータ優先度データOPPRIO<sub>m</sub>をチェックし（ステップS82）、この両チェック結果に応じて、当該演算対象オペレータmに対するオペレータ演算処理を実行するか否かを判別する（ステップS83）。

## 【0110】

ステップS83で、演算対象オペレータmに対するオペレータ演算処理を実行するときには、チャンネルnは前フレームから継続して発音されているか否かを



判別し（ステップS84）、チャンネルnは前フレームから継続して発音されているときには、音色レジスタTONEPARnのオペレータデータOPmDATAn内のバッファOPBUFmに格納された各データに基づいて、前フレーム演算終了時のオペレータmの状態にオペレータデータOPmDATAnを復帰させる（ステップS85）。これは、各オペレータデータOPmDATAn内のバッファOPBUFmには、その直前に演算された演算結果が保存されているため、この保存結果を用いることにより、当該直前のオペレータデータOPmDATAnの状態に復帰させることができるからである。そして、このようにオペレータデータOPmDATAnを前フレームの演算終了時の状態に復帰させるのは、今回のフレームにおけるチャンネルnの楽音波形データは、前フレームから継続するものとして生成する必要があるからである。

## 【0111】

一方、ステップS84で、チャンネルnは前フレームから継続して発音されていないときには、ステップS85をスキップしてステップS86に進む。

## 【0112】

ステップS86では、図17および18を用いて後述する演算対象オペレータmについてのオペレータ演算処理サブルーチンを実行する。

## 【0113】

続くステップS87では、変数mの値を“1”だけインクリメントして更新し、ステップS88では、関連オペレータすべてについて、すなわちアルゴリズム指定データALGORnにより示されるオペレータデータOPmDATAnすべてについてオペレータ演算を完了したか否かを判別する（ステップS88）。

## 【0114】

ステップS88で、関連オペレータについての演算が残っているときには前記ステップS82に戻り、前述の処理を繰り返す一方、関連オペレータすべてについて演算を完了したときには、本チャンネルnについてのFM演算処理を終了する。

## 【0115】

なお、ステップS82およびS83では、CPU3の負荷状態をチェックして

演算対象オペレータ $m$ に対するオペレータ演算を実行するか否かを判別するようにしたが、これに限らず、CPU3の負荷状態に拘わらず、優先度の低いオペレータに対するオペレータ演算を実行しないようにしてもよい。これにより、CPU3の能力があまり高くない場合であっても、発音数を増大させることができる。

【0116】

図17および18は、前記ステップS86の演算対象オペレータ $m$ についてのオペレータ演算処理サブルーチンの詳細な手順を示すフローチャートであり、図19は、このオペレータ演算処理でなされるオペレータ演算の基本的な流れを示す図である。以下、図19も参照しながら、演算対象オペレータ $m$ についてのオペレータ演算処理を説明する。

【0117】

図17において、まず、演算対象オペレータ $m$ のオペレータデータ $OPmDATA_n$ 内のオペレータオンパラメータ $OPONm$ がオン（“1”）であるか否かを判別し（ステップS91）、 $OPONm=0$ のとき、すなわち演算対象オペレータ $m$ がオペレータ演算を行う必要のないものであるときには、直ちに本オペレータ演算処理を終了する一方、 $OPONm=1$ のとき、すなわち演算対象オペレータ $m$ がオペレータ演算を行うべきものであるときにはステップS92に進む。

【0118】

ステップS92では、オペレータデータ $OPmDATA_n$ 内の前記サンプリング周波数指定データ $FSAMPm$ が“0”であるか否か、すなわち標準モードのサンプリング周波数 $FSMAX$ で楽音波形を生成するか否かを判別し、 $FSAMPm=0$ のときには、各オペレータでの演算処理は標準モードのサンプリング周波数で楽音波形生成を行うことが基本とされているため、オペレータデータ $OPmDATA_n$ 内の前記エンベロープパラメータ $EGPARm$ の設定値に応じて $EGm$ 演算を行い、その演算結果を前記EG状態バッファ $EGSTATEm$ に格納する（ステップS93）。

【0119】

一方、ステップS92で、 $FSAMPm \neq 0$ 、たとえば $FSAMPm=f$ のと

き、すなわち標準モードのサンプリング周波数  $F_{SMA X}$  を  $2^{-f}$  倍に変更し、変更後の周波数で楽音波形生成を行うときには、エンベロープパラメータ  $E G P A R m$  中、レートが変化するパラメータ（以下、「変化レート関連パラメータ」という）を  $2^f$  倍にして  $A E G$  演算を行い、その演算結果を  $E G$  状態バッファ  $E G S T A T E m$  に格納する。このように、変化レート関連パラメータのレートを  $2^f$  倍にしてエンベロープ生成演算を行うようにしたのは、このとき、サンプリング周波数は  $F_{SMA X} \times 2^{-f}$  に低減されるため、これに応じて、本実施の形態では、エンベロープパラメータ  $E G P A R m$  の変化レート関連パラメータのレート、すなわち時間変化を速くして当該サンプリング周波数における楽音波形生成を行い、その後、生成された波形サンプルを  $2^f$  個の対応する連続バッファに書き込むことにより、元の音高の楽音となるように調整する方法を採っているからである。

【0120】

このように、ステップ  $S 9 3$  または  $S 9 4$  は、図 19 におけるエンベロープデータ  $A E G m$  の算出を行っている。

【0121】

続くステップ  $S 9 5$  では、 $A E G m$  演算により算出されたデータ  $A E G m$  にオペレータデータ  $O P m D A T A n$  内の前記トータルレベルパラメータ  $T L m$  の値を乗算し、図 19 に示すように、演算対象オペレータ  $m$  の出力レベル  $A M P m$  ( $= A E G m \times T L m$ ) を算出する。

【0122】

そして、ステップ  $S 9 3$  または  $S 9 4$  で算出された振幅制御用エンベロープデータ  $A E G m$  およびステップ  $S 9 5$  で算出された演算対象オペレータ  $m$  の出力レベル  $A M P m$  をそれぞれチェックし（ステップ  $S 9 6$ ）、各データ値  $A E G m$  および  $A M P m$  がそれぞれ、たとえば所定時間所定レベル以下になり、演算対象オペレータ  $m$  の出力をなしとしてよいか、すなわち演算対象オペレータ  $m$  における楽音波形演算を終了してよいか否かを判別し（ステップ  $S 9 7$ ）、その答えが肯定（YES）のときにはステップ  $S 9 8$  に進む一方、その答えが否定（NO）のときには図 18 のステップ  $S 1 0 1$  に進む。

## 【0123】

ステップS98では、演算対象オペレータmは前記「キャリア」であるか否かを判別し、「キャリア」のときにはステップS99に進み、演算対象オペレータmおよび演算対象オペレータmのみを変調している「モジュレータ」の各バッファOPBUFをクリアし、波形演算を停止させた後に、本オペレータ演算処理を終了する。このように、演算対象オペレータmが「キャリア」である場合に、演算対象オペレータmのみでなく、演算対象オペレータmのみを変調しているモジュレータの波形演算を停止させるようにしたのは、キャリアは、前記図4に示すように、楽音波形データを最終的に出力するオペレータであるため、キャリアからの出力をなしとしてよいとき、すなわちステップS97からS98を経てS99へ移行するときには、その前段であるモジュレータの出力もなしとしてよいからである。ただし、当該モジュレータが他のキャリアを変調しているときには、当該モジュレータの波形演算を停止させることはできないため、これを「のみ」で担保している。

## 【0124】

一方、ステップS98で、演算対象オペレータmがキャリアでないとき、すなわちモジュレータのときには、演算対象オペレータmのバッファOPBUFmのみをクリアしてその波形演算を停止させた（ステップS100）後に、本オペレータ演算処理を終了する。

## 【0125】

図18のステップS101では、アルゴリズム指定データALGORnをチェックし、次いでステップS102では、演算対象オペレータmが他のオペレータから変調を受けているか否かを判別する。

## 【0126】

ステップS102で、演算対象オペレータmが他のオペレータから変調を受けているときには、当該変調を行っている各オペレータデータOPkDATAn内の前記オペレータ出力値バッファOPOUTkに格納されているオペレータ出力データをそれぞれ加算し、演算対象オペレータmの前記変調データ入力バッファMODINmに格納する（ステップS103）一方、演算対象オペレータmが他

のオペレータから変調を受けていないときには、ステップS103をスキップしてステップS104に進む。

【0127】

ステップS104では、前記ステップS92と同様にして、オペレータデータOPmDATAn内の前記サンプリング周波数指定データFSAMPmが“0”であるか否かを判別し、FSAMPm=0のときにはステップS105に進む一方、FSAMPm≠0のときにはステップS110に進む。

【0128】

ステップS105では、位相値を更新する位相値更新演算を行い、その演算結果を、演算対象オペレータmのオペレータデータOPmDATAn内の位相値バッファPHBUFm（以下、この内容を「位相値PHBUFm」という）に格納する。ここで、位相値更新演算とは、図19の破線Aで囲まれた演算、すなわち次式で示す演算をいう。

【0129】

$$\text{MODINm} + \text{FBm} + \text{FNO}n \times \text{MULTm} + \text{PHBUFm}$$

ただし、MODINm、FBmは、それぞれオペレータデータOPmDATAn内の変調データ入力バッファMODINmおよびフィードバック出力値バッファFBmに格納された値を示し、FNO<sub>n</sub>は、楽音パラメータVOICE<sub>n</sub>内の周波数ナンバFNO<sub>n</sub>を示し、MULTmは、オペレータデータOPmDATAn内の前記周波数倍数データMULTmを示し、PHBUFmは、オペレータデータOPmDATAn内の位相値バッファPHBUFmに格納された値の前回値を示している。

【0130】

続くステップS106では、ステップS105で算出された位相値PHBUFmに基づいてテーブルアドレスを算出し、演算対象オペレータmのウェーブセレクトデータWSELmに応じて選択された基本波形（たとえば、前記図5で示した8種類の基本波形から選択された波形）データ（以下、「基本波形テーブル」という）から、この算出アドレスが示す位置のデータWAVEm（PHBUFm）を読み出し、このデータWAVEm（PHBUFm）に、前記ステップS95

で算出した出力レベルAMP<sub>m</sub>を乗算して、演算対象オペレータ<sub>m</sub>のオペレータ出力値バッファOPOUT<sub>m</sub> (=WAVE<sub>m</sub> (PHBUF<sub>m</sub>) ×AMP<sub>m</sub>) に格納する。

【0131】

そして、ステップS107では、次式によりフィードバックサンプル演算を行い、その演算結果を演算対象オペレータ<sub>m</sub>のフィードバック出力値バッファFB<sub>m</sub>に格納する。

【0132】

$$0.5 \times (FB_m + OPOUT_m \times FBL_m)$$

ただし、OPOUT<sub>m</sub>は、ステップS106で発生させた波形サンプルデータを示し、FBL<sub>m</sub>は、演算対象オペレータ<sub>m</sub>の前記フィードバックレベルデータFBL<sub>m</sub>を示している。ここで、フィードバックサンプル演算を行うのは、寄生発振を防止するためである（後述するステップS112も同様）。

【0133】

さらに、ステップS108では、前記ステップS98と同様にして、演算対象オペレータ<sub>m</sub>が「キャリア」であるか否かを判別し、「モジュレータ」のときには、直ちに本オペレータ演算処理を終了する一方、「キャリア」のときには、ステップS106で発生させた波形サンプルデータOPOUT<sub>m</sub>に、楽音パラメータVOICE<sub>n</sub>の前記音量データVOL<sub>n</sub>を乗算し、その乗算結果 (=OPOUT<sub>m</sub> ×VOL<sub>n</sub>) を、前記波形バッファ（対応バッファ）における今回の書き込み位置を示すポインタが指示する位置に加算するとともに、当該ポインタの値を“1”だけ進めて更新した（ステップS109）後に、本オペレータ演算処理を終了する。

【0134】

ステップS110では、位相値更新演算を行い、その演算結果を位相値バッファPHBUF<sub>m</sub>に格納する。このステップS110の演算処理は、ステップS105の演算処理に対して、図19のブロックBで示す処理を追加した点が異なるのみである。これは、FSAMP<sub>m</sub> = f (≠0) であるため、位相値をfビットだけシフトアップして、すなわち位相値バッファPHBUF<sub>m</sub>の値を2<sup>f</sup>倍して

、前記基本波形テーブルの読み出しアドレスをサンプリング周波数  $F_{SMA} \times 2^{-f}$  時のアドレスに変更する必要がある、すなわち飛ばし読みする必要があるからである。

【0135】

次に、前記ステップS106と同様にして、次式により波形サンプルを発生させ、オペレータ出力値バッファ  $OPOUT_m$  に格納する。

【0136】

$$WAVE_m (2^f \times PHBUF_m) \times AMP_m$$

そして、前記ステップS107と同様にして、フィードバックサンプル演算を実行する（ステップS112）。

【0137】

次いで、ステップS113で、前記ステップS108と同様にして、演算対象オペレータ  $m$  が「キャリア」であるか否かを判別し、「モジュレータ」のときには、直ちに本オペレータ演算処理を終了する一方、「キャリア」のときには、前記ステップS109と同様にして、ステップS111で発生させた波形サンプルデータ  $OPOUT_m$  に、楽音パラメータ  $VOICE_n$  の前記音量データ  $VOL_n$  を乗算し、その乗算結果（ $= OPOUT_m \times VOL_n$ ）を、前記波形バッファ中、前記ポインタが示す位置から連続する  $2^f$  個の領域（バッファ）に加算するとともに、そのポインタの値を“ $2^f$ ”だけ進めて更新した（ステップS109）後に、本オペレータ演算処理を終了する。なお、ステップS109で、同じ値のサンプルデータを書き込む際に、必要に応じてサンプル間の補間演算を行い、その補間値を書き込むようにしてもよい。

【0138】

なお、本実施の形態では、ステップS106およびS111で説明したように、基本波形データとしては基本波形テーブルに格納された値を使用するようにしたが、これに限らず、基本波形データを演算により生成するようにしてもよい。もちろん、テーブルデータと演算との組合せにより、基本波形データを生成するようにしてもよい。

【0139】

また、ステップS106およびS110で、基本波形テーブルを読み出すときのアドレスは、それぞれステップS105およびS110で算出された位相値PHBUFmに基づいて算出されたアドレスを使用した。これに限らず、この位相値PHBUFmを演算や非線形特性テーブル等で歪ませたものを使用するようにしてもよい。

#### 【0140】

図20は、前記図10のステップS19の音色設定処理サブルーチンの詳細な手順を示すフローチャートである。

#### 【0141】

同図において、まず、MIDIチャンネルとその各チャンネルに対応する音色の設定を行う（ステップS121）。前述したように、本実施の形態では、MIDIチャンネルとその対応音色はMIDI-CHボイステーブルで決定され、このMIDI-CHボイステーブルにロードされるテーブルデータはハードディスク等に格納され、ユーザが選択したMIDIファイルがロードされると同時に、その対応するテーブルデータがMIDI-CHボイステーブルにロードされるので、ステップS121の処理は、現在ロードされているテーブルデータの編集または新たなテーブルデータのロード（読み込み）である。

#### 【0142】

なお、各MIDIチャンネル毎にそれぞれ希望のオペレータ数をユーザが設定可能とし、MIDI-CHボイステーブル中の音色番号を変更するときに、当該チャンネルに希望オペレータ数が設定されている場合には、そのオペレータ数に一致するまたはそのオペレータ数以内の楽音パラメータVOICEに対応する音色番号を一覧表示し、その中からユーザが所望の音色番号を選択して設定するようにしてもよい。このとき、CPU3の負荷状態に応じて、当該チャンネルに設定された希望オペレータ数を自動的に変更し、その変更後のオペレータ数以内の音色番号を一覧表示するようにしてもよい。さらに、ユーザがMIDI-CHボイステーブル中のあるチャンネルの音色番号を変更したときに、その音色番号に対応する楽音パラメータVOICEの構成オペレータ総数をチェックし、CPU3の負荷状態に応じて、当該チャンネルにこの音色を割り当てできない等の警告



を表示するようにしてもよい。この警告に加えて、当該チャンネルの音色番号を、オペレータ数の小さい代替音色の音色番号に自動的に変更するようにしてもよい。

【0143】

以上説明したように、本実施の形態では、CPU3の能力や動作環境、目的および設定等に応じて、FM演算処理に用いるオペレータ数をフレキシブルに変更するようにしたので、CPU3の負荷の軽重および出力楽音波形の品質の高低を自由に変更することができ、これにより、音源システム全体の自由度を向上させることができる。

【0144】

なお、本実施の形態では、楽音生成処理としてFM音源方式を用いて説明したが、これに限らず、たとえば、楽音波形生成ブロックを組み合わせることでAM（振幅変調）やPM（位相変調）など所定の信号処理を行うような音源にも容易に適用できる。また、本発明に含まれるCPUの負荷の軽減方法は波形メモリ読み出し方式の音源、物理モデル音源をソフトウェアで実現する際にも適用できるものである。

【0145】

また、本実施の形態では、パソコンでの応用を主な例としたが、ゲームやカラオケなどのアミューズメント機器、電子楽器、一般電子機器への適用も容易に可能である。また、パソコンのオプションとしての音源ボードや音源ユニットに適用することも可能である。

【0146】

なお、本発明によるソフトウェアは、フロッピーディスクや光磁気ディスク、CD-ROMなどのディスクメディア、あるいはメモリカードなどで供給するようにもできる。また、ソフトウェアデータを記憶した半導体メモリチップ（典型的にはROM）をコンピュータ機器に差し替え、追加するようにしてもよい。さらには通信I/F14を通して、本発明に係る音源ソフトウェアを配信するようにしてもよい。

【0147】

また、本発明に係る音源ソフトウェアが、コンピュータシステムの中で、どういう位置づけ（アプリケーション・ソフトとするか、あるいは、たとえばデバイスドライバ・ソフトにするか）で起動、動作するかはシステム構成やOSに応じて適宜、決めればよい。

【0148】

また、本発明に係る音源ソフトウェアまたはその機能を、他のソフトウェア、たとえば、ゲームなどのアミューズメントソフト、カラオケソフトや自動演奏・伴奏ソフトなどに組み込んでもよい。さらには、コンピュータOSに直接、組み込んでしまってもよい。

【0149】

【発明の効果】

以上説明したように、本発明に依れば、選択された発音チャンネルに、設定されたアルゴリズムが示す楽音波形生成ブロックが割り当てられ、該割り当てられた楽音波形生成ブロックが当該アルゴリズムが示す組み合わせ態様で組み合わせられて楽音波形生成演算が行われ、楽音波形データが生成されるので、発音割り当てされる前に、各発音チャンネルに対する楽音波形生成ブロックの個数を任意に変更することができ、これにより、楽音波形データ生成手段の能力に応じて、該楽音波形データ生成手段に対する負荷状態および生成される楽音波形データの品質をフレキシブルに変更することが可能となる効果を奏する。

【0150】

また、選択された発音チャンネルに、当該生成すべき楽音波形データの音色に対応して設定されたアルゴリズムが示す楽音波形生成ブロックが割り当てられ、該割り当てられた楽音波形生成ブロックが当該アルゴリズムが示す組み合わせ態様で組み合わせられて楽音波形生成演算が行われ、楽音波形データが生成されるので、上記効果と同様の効果を奏することができる。

【0151】

好ましくは、音色設定手段により音色が設定されるときに、当該演奏パートにブロック数設定手段により楽音波形生成ブロック数が設定されている場合には、当該演奏パートに設定された音色が、該設定ブロック数の範囲内の楽音波形生成

ブロックを備えた音色に変更されるので、上記効果をさらに高めることができる。

#### 【0152】

また、好ましくは、前記発音チャンネルにおける楽音波形生成演算中に、所定の条件に応じて、当該発音チャンネルに割り当てられた楽音波形生成ブロックの個数に変更されるので、発音中に、楽音波形データ生成手段の能力に応じて、該楽音波形データ生成手段に対する負荷状態および生成される楽音波形データの品質をフレキシブルに変更することができる。

#### 【0153】

さらに、本発明によれば、パソコンなど、音楽演奏の他にも複数任意のタスク処理（たとえばワープロ、ネットワーク通信など）を実行させる場合が多いコンピュータ機器では、ソフト音源処理実行中に音源以外のタスクにCPUパワーがかかっても、楽音の途切れなどの障害の発生を少なくできる。これは、見方を変えれば、音源処理を実行中にも、より多くのタスクが処理できるということができる。

#### 【図面の簡単な説明】

##### 【図1】

本発明の実施の一形態に係る音源システムが備えたソフトウェアのシステム構成の一例を示す図である。

##### 【図2】

本発明の実施の一形態に係る音源システムの概略構成を示す図である。

##### 【図3】

図1の音源システムが行う楽音生成処理の概要を説明するための図である。

##### 【図4】

FM音源方式による楽音生成処理の概要を説明するための図である。

##### 【図5】

基本波形テーブルとして選択される元となる基本波形データの一例を示す図である。

##### 【図6】

割り当てられた発音チャンネルで発音すべき楽音の音色パラメータを展開する音色レジスタ群を示す図である。

【図 7】

楽音パラメータ VOICE<sub>j</sub> のデータフォーマットを示す図である。

【図 8】

各 MIDI チャンネル (CH) 毎に選択設定された楽音パラメータ VOICE<sub>n</sub> の音色番号を記憶する MIDI-CH ボイステーブルを示す図である。

【図 9】

図 1 の音源システムの CPU が実行する初期プログラムの手順を示すフローチャートである。

【図 10】

図 9 の初期プログラムに続いて CPU が実行するメインプログラムの手順を示すフローチャートである。

【図 11】

図 10 の MIDI 処理サブルーチンの詳細な手順を示すフローチャートである。

【図 12】

図 11 の MIDI 処理サブルーチンの続きを示すフローチャートである。

【図 13】

CH シーケンスレジスタのフォーマットの一例を示す図である。

【図 14】

図 10 の波形演算処理サブルーチンの詳細な手順を示すフローチャートである。

【図 15】

図 14 の波形演算処理サブルーチンの続きを示すフローチャートである。

【図 16】

図 15 のチャンネル  $n$  のついて FM 演算処理サブルーチンの詳細な手順を示すフローチャートである。

【図 17】

図16の演算対象オペレータmについてのオペレータ演算処理サブルーチンの詳細な手順を示すフローチャートである。

【図18】

図17の演算対象オペレータmについてのオペレータ演算処理サブルーチンの続きを示すフローチャートである。

【図19】

図17および18のオペレータ演算処理でなされるオペレータ演算の基本的な流れを示す図である。

【図20】

図10の音色設定処理サブルーチンの詳細な手順を示すフローチャートである。

【符号の説明】

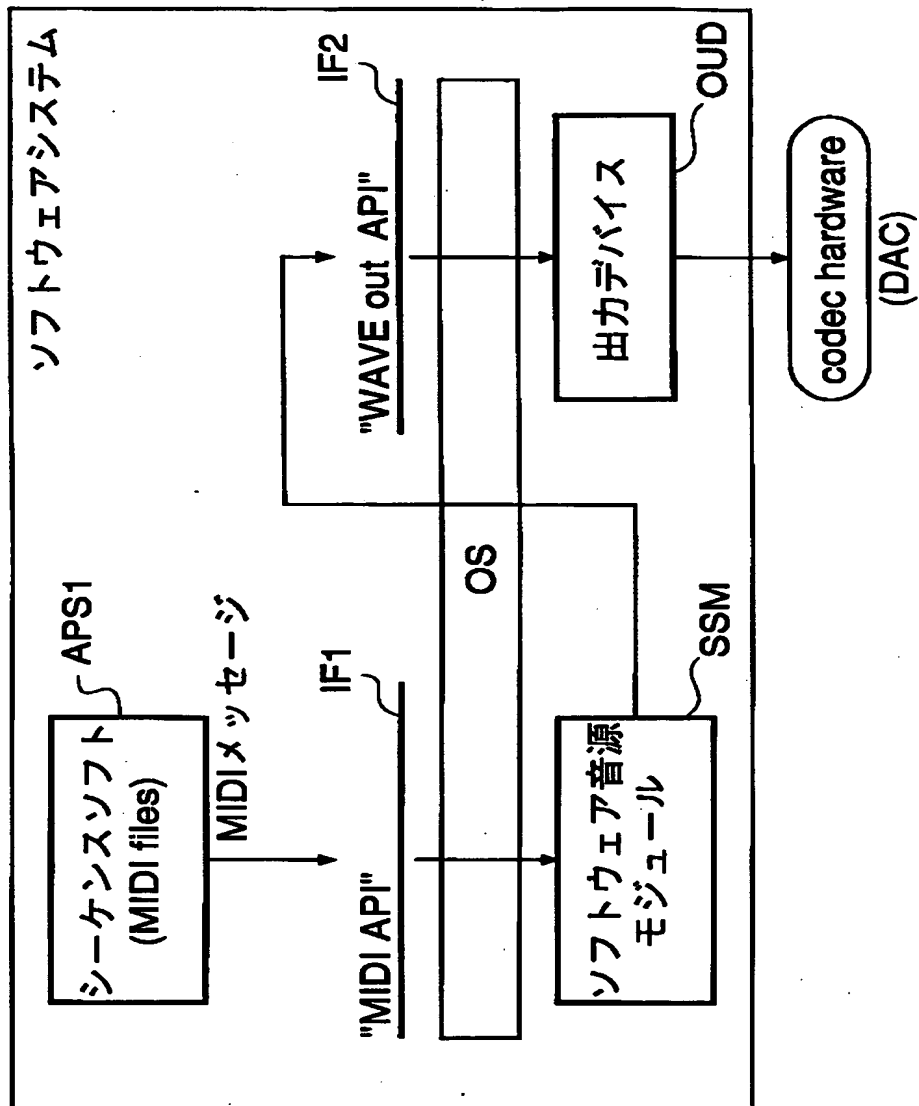
3 CPU (アルゴリズム設定手段、発音チャンネル選択手段、楽音波形データ生成手段、音色設定手段、ブロック数設定手段、音色変更手段、ブロック数変更手段、削減態様設定手段、サンプリング周波数設定手段)

OPmDATAn オペレータデータ (楽音波形生成ブロック)

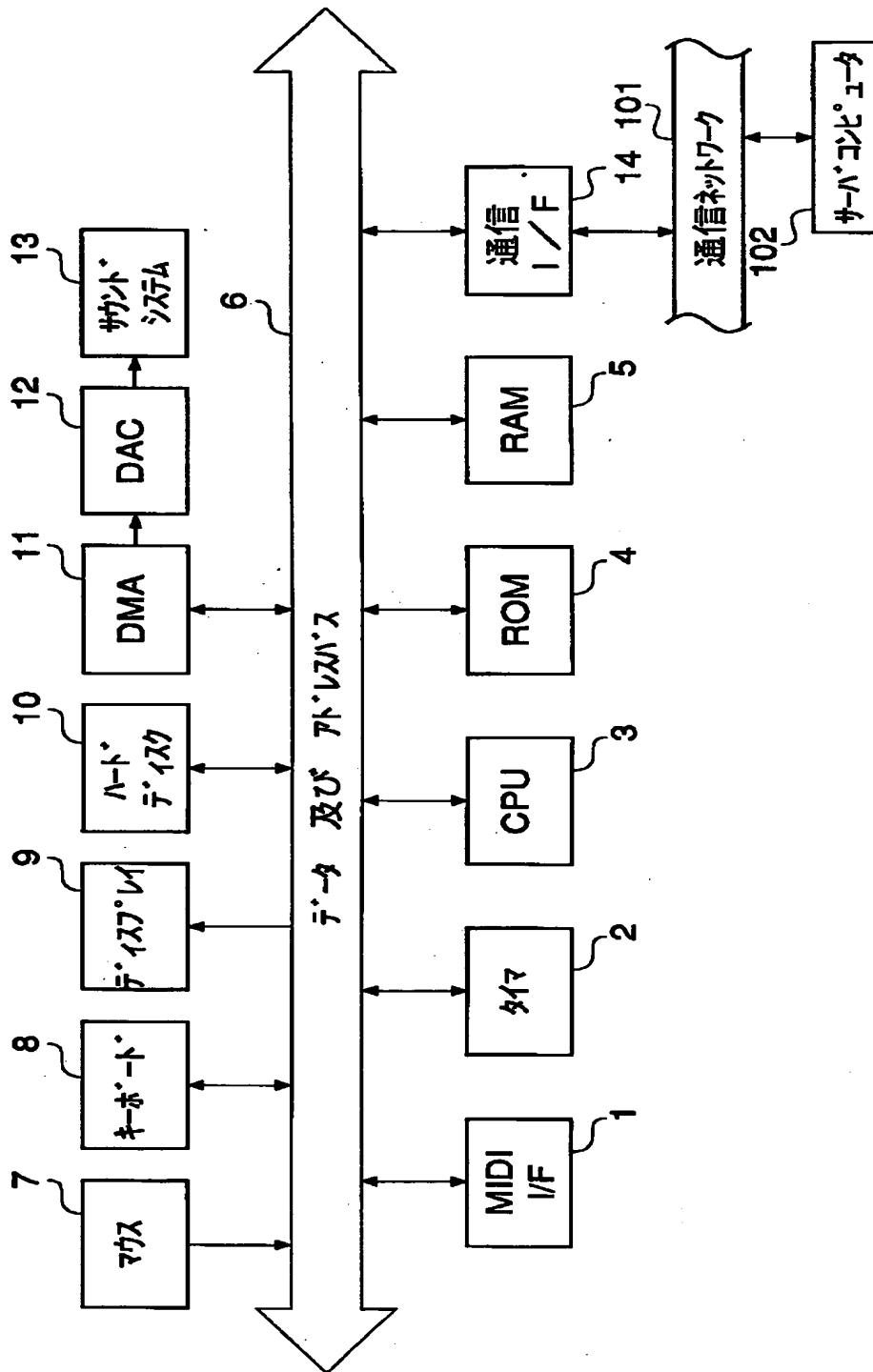
SSM ソフトウェア音源モジュール (楽音波形生成ブロック、アルゴリズム設定手段、発音チャンネル選択手段、楽音波形データ生成手段、音色設定手段、ブロック数設定手段、音色変更手段、ブロック数変更手段、削減態様設定手段、サンプリング周波数設定手段)

【書類名】 図面

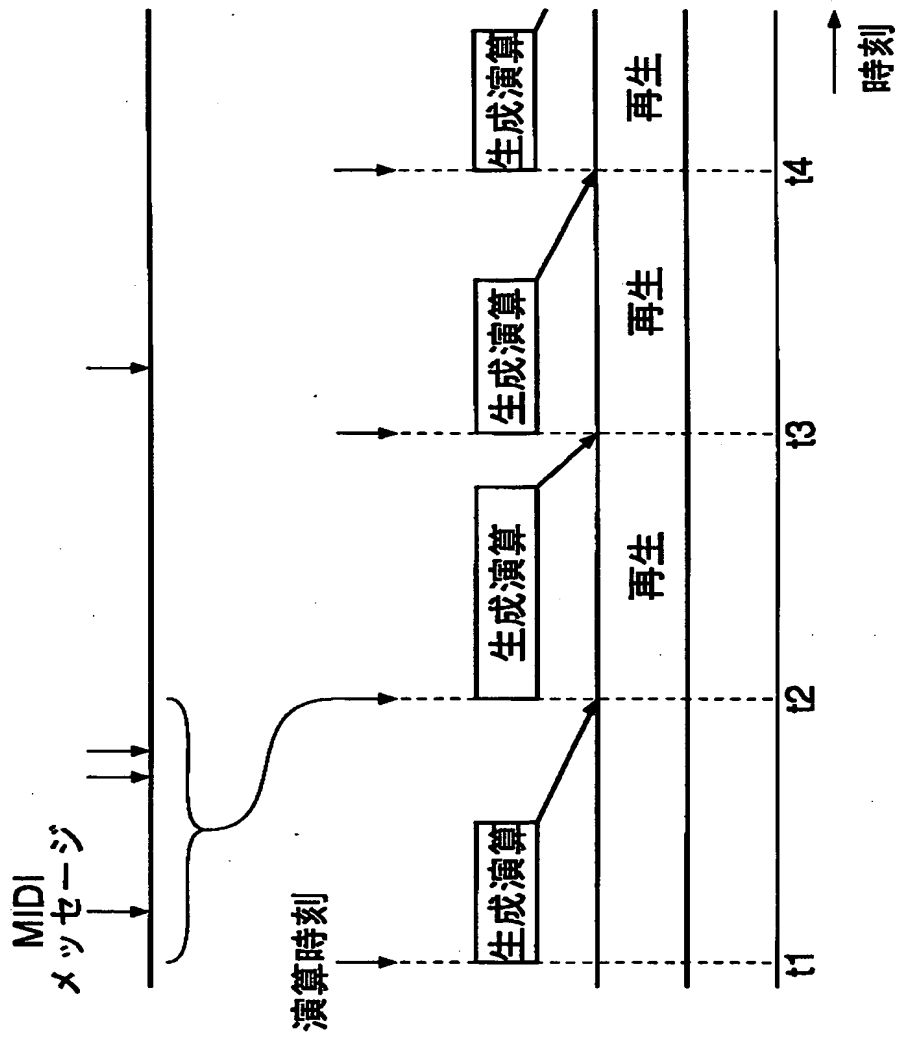
【図1】



【図2】

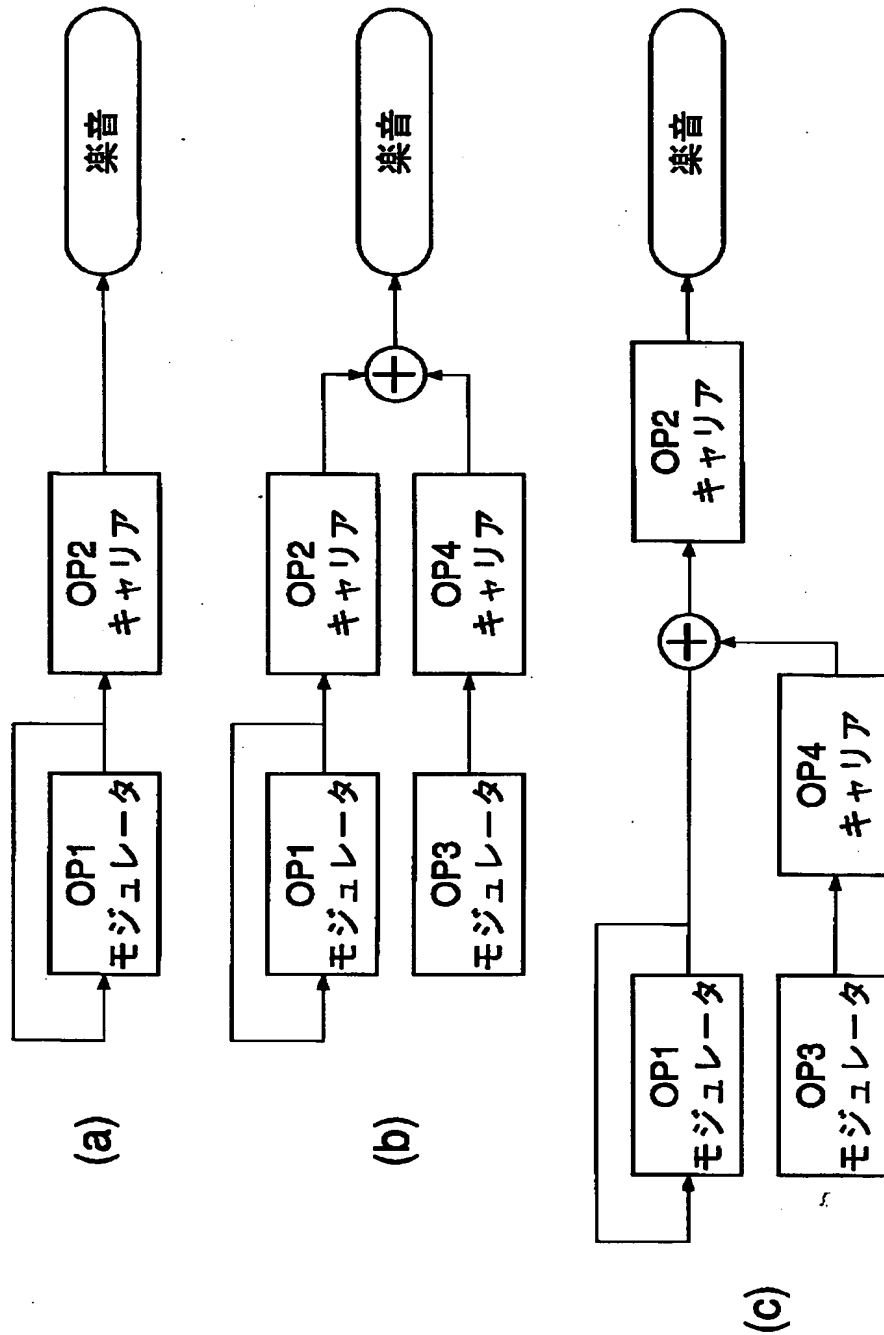


【図3】


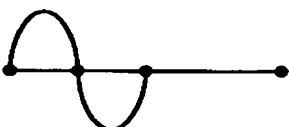
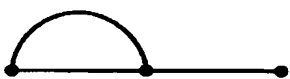


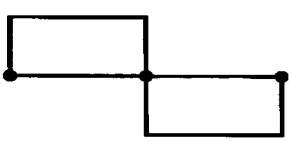

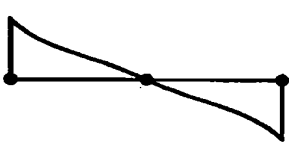




【図4】



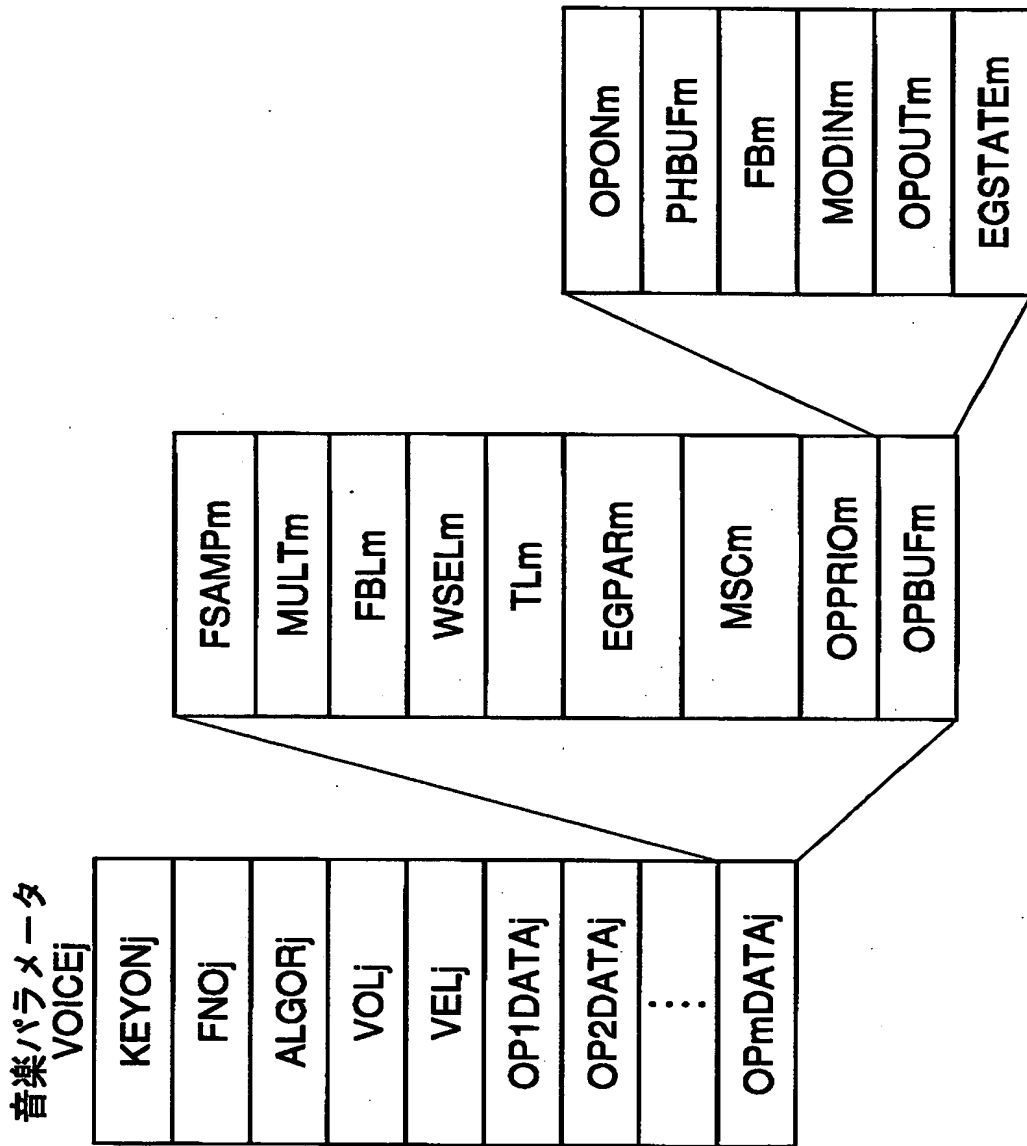
【図5】

WS=0		WS=4	
WS=1		WS=5	
WS=2		WS=6	
WS=3		WS=7	

【図6】

TONEPAR1
TONEPAR2
⋮
TONEPAR <sub>n</sub>
⋮

【図 7】

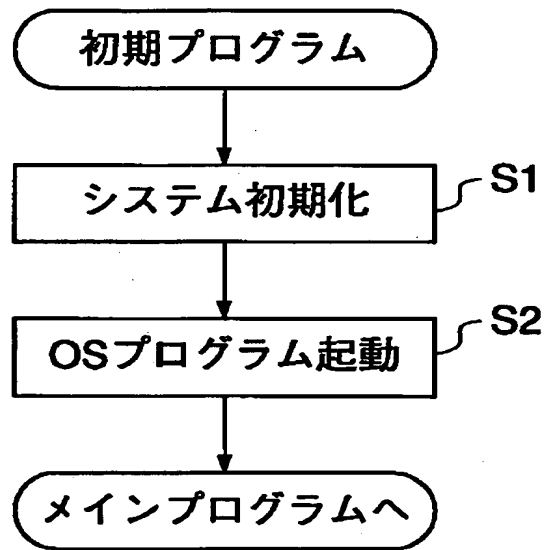


【図8】

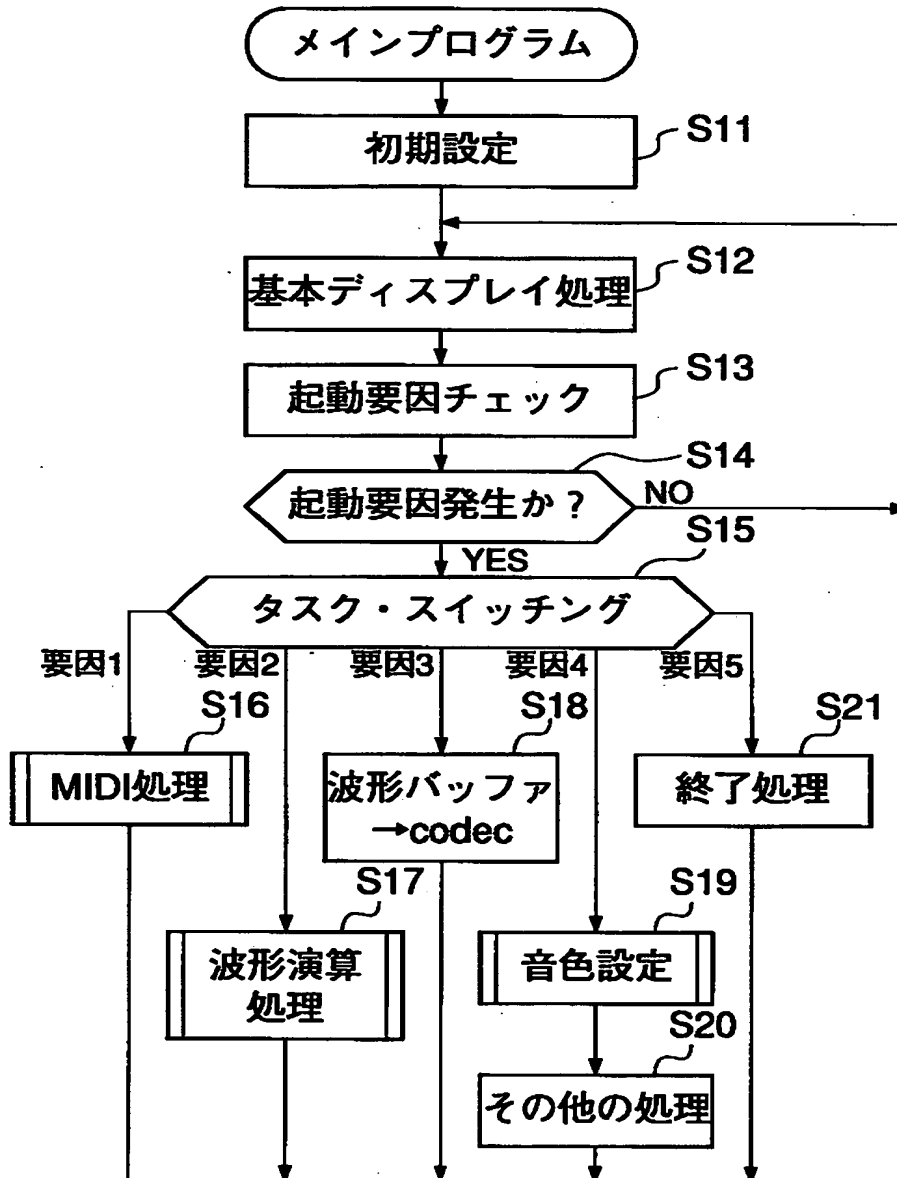
MIDI-CH  
ボイステーブル

VOICENO1
VOICENO2
⋮
VOICENO16

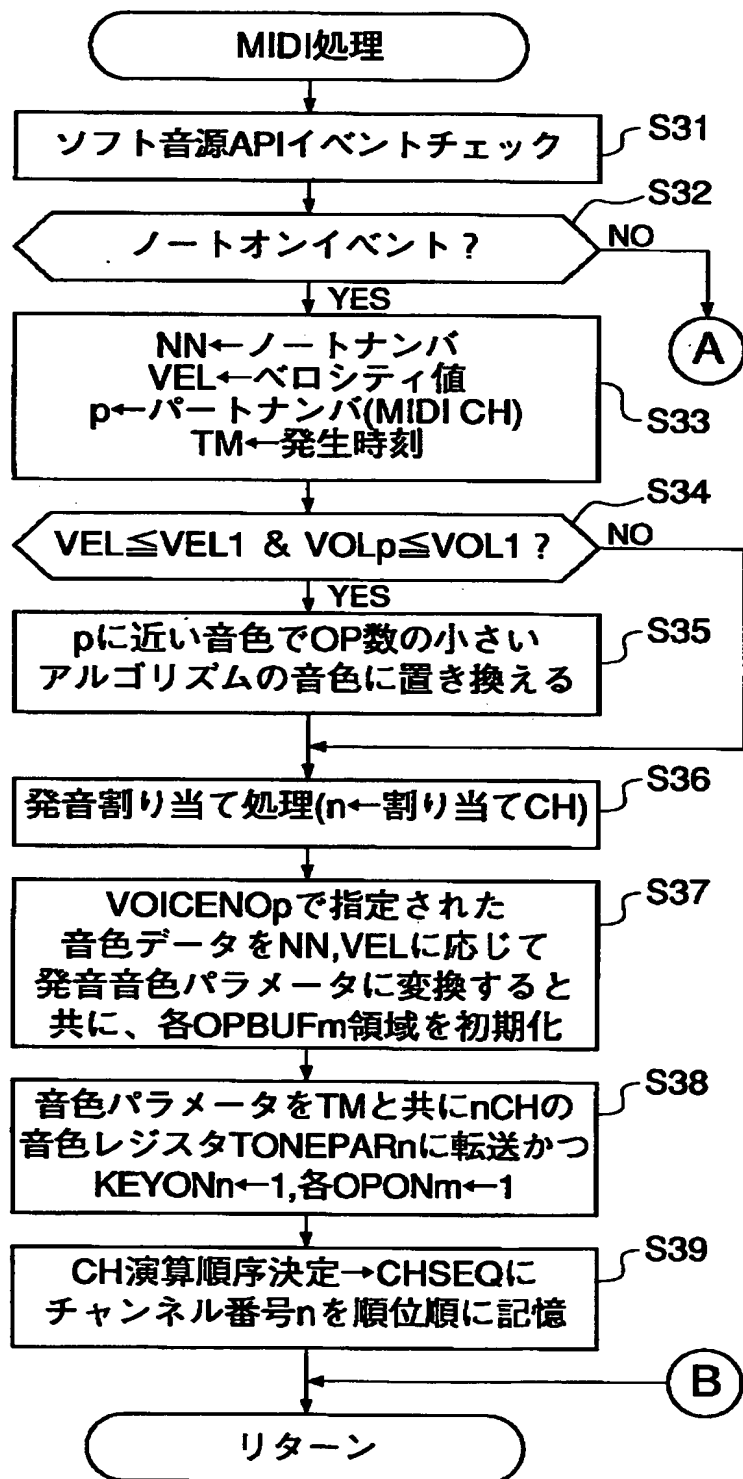
【図9】



【図10】

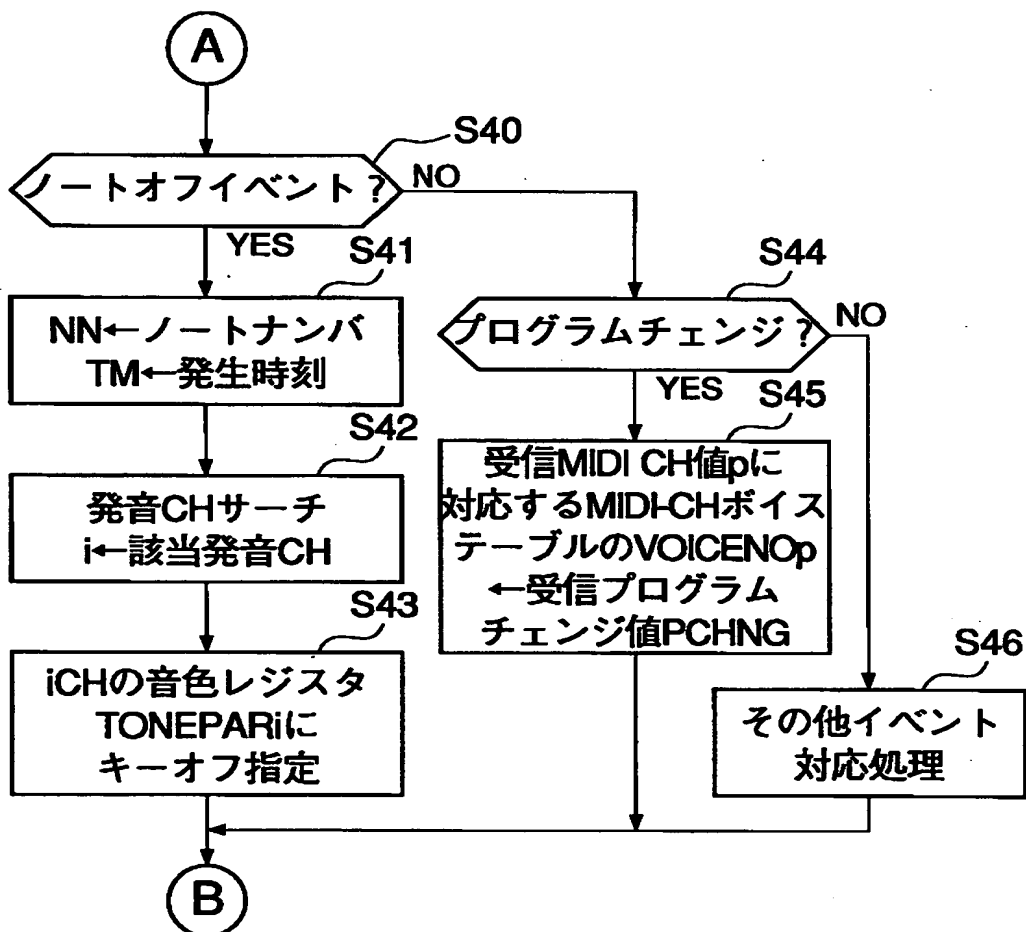


【図11】





【図12】

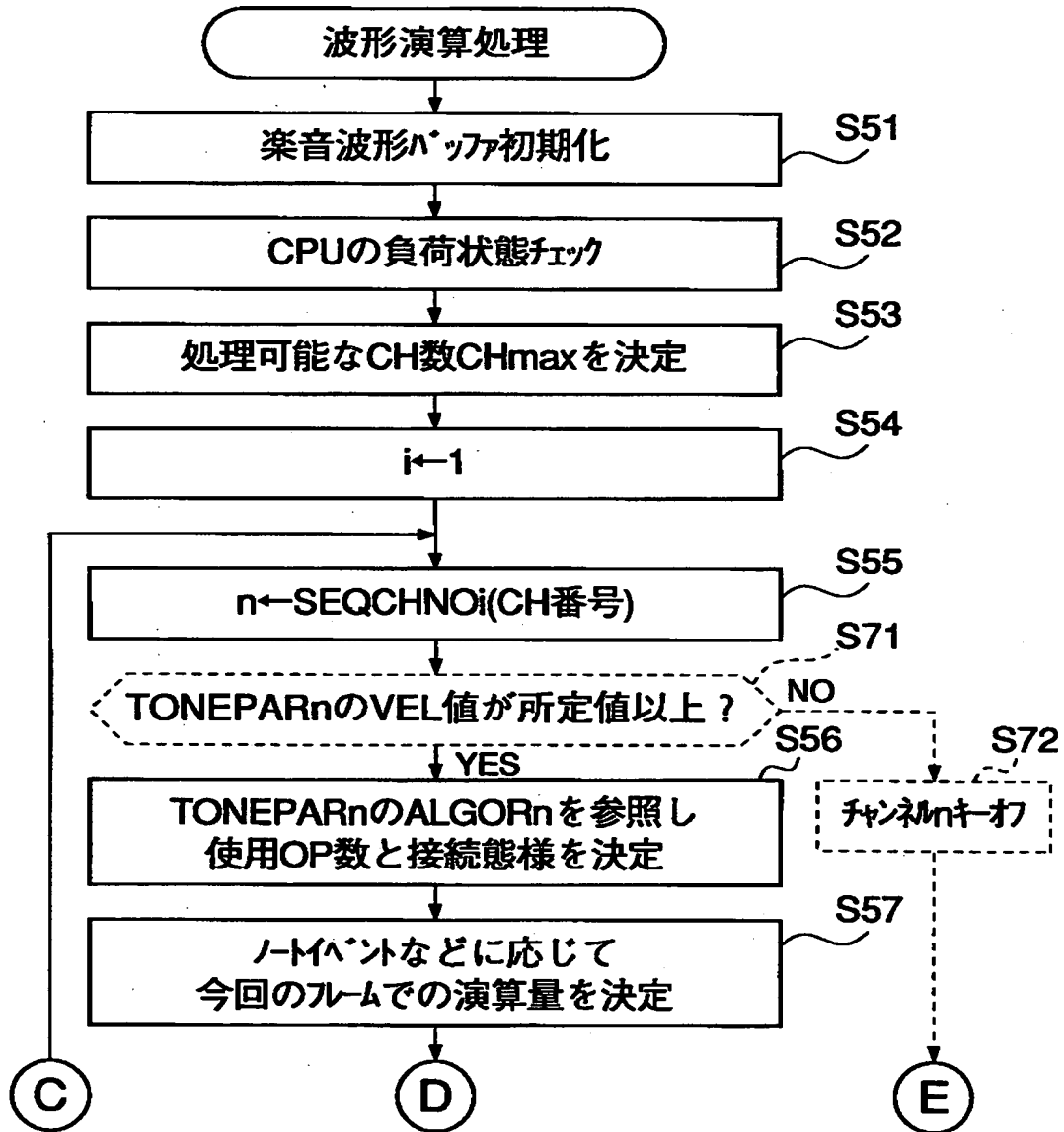


【図13】

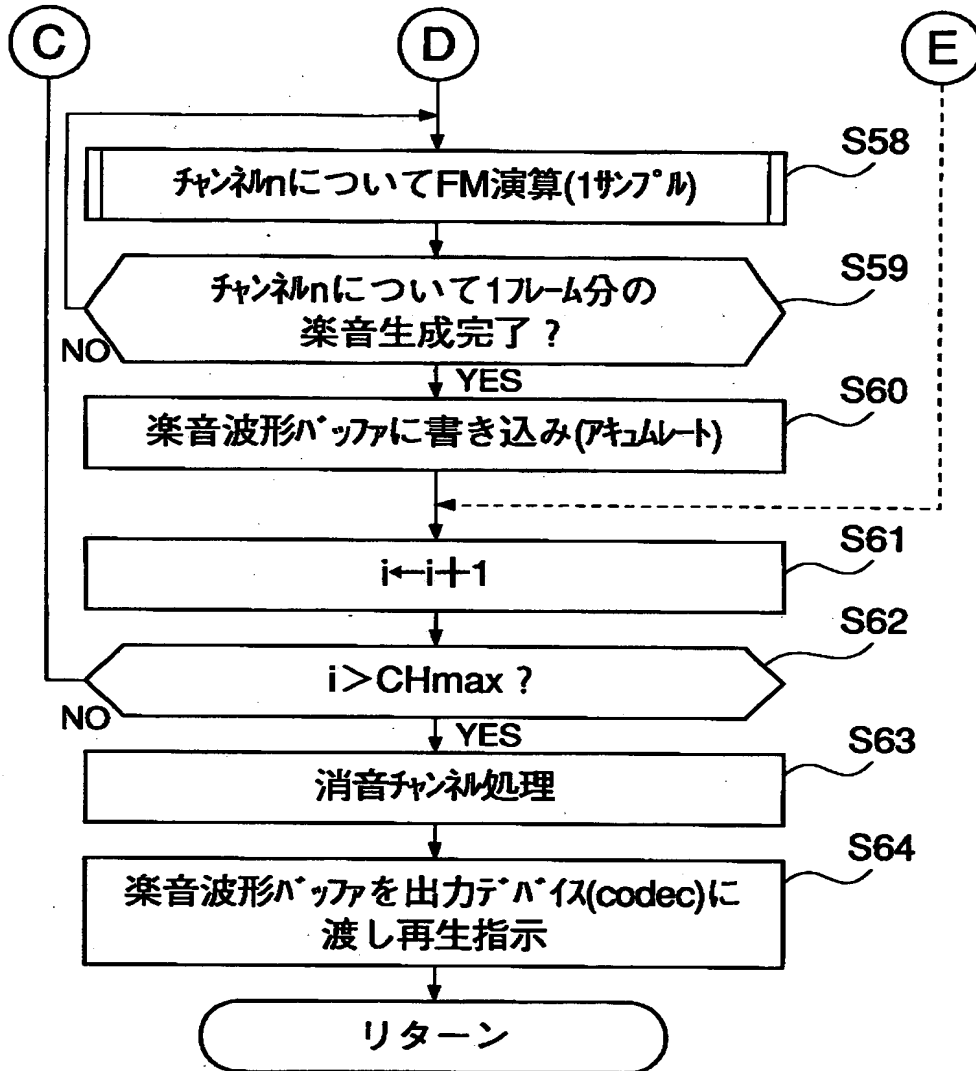
CHシーケンス  
レジスタ

SEQCH1
SEQCH2
⋮
SEQCHn

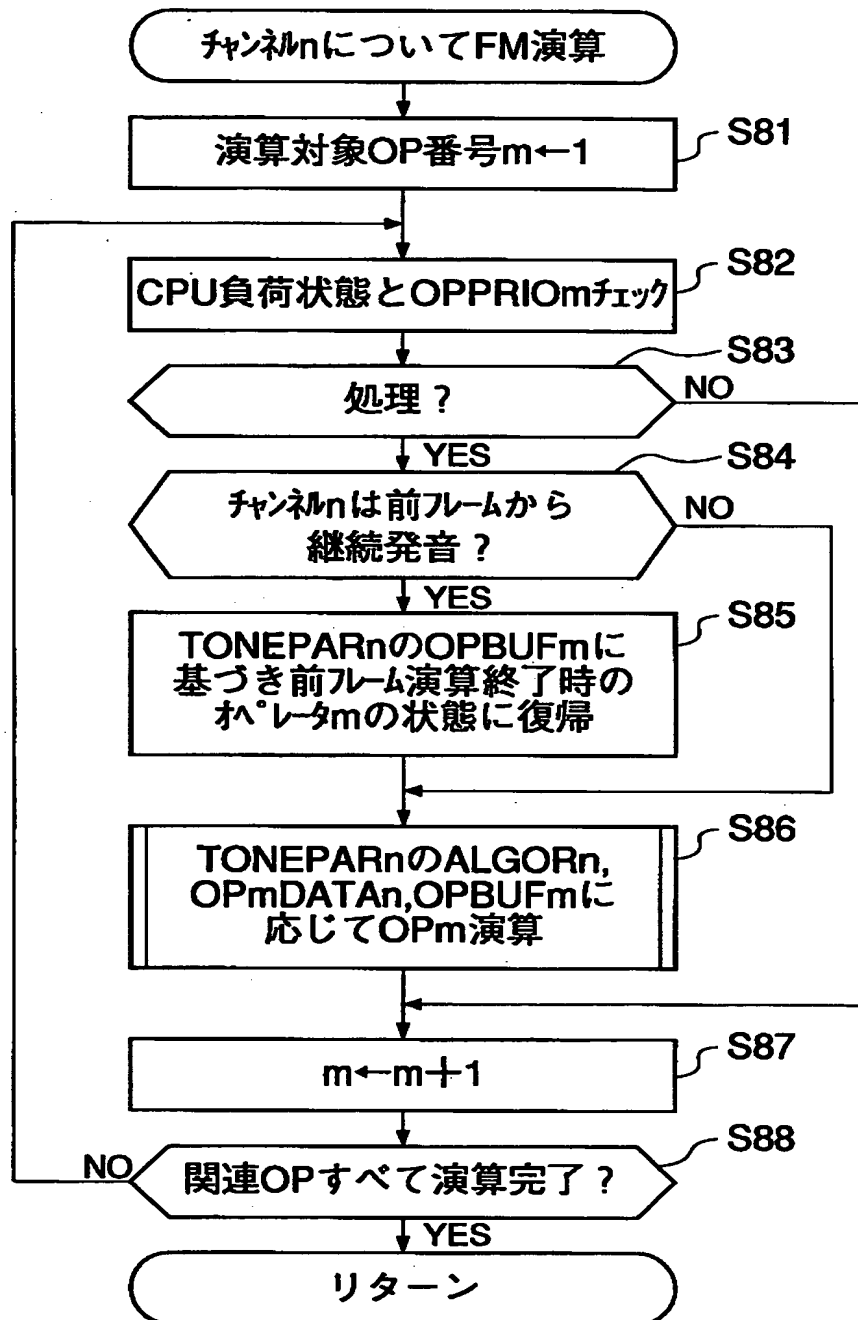
【図14】



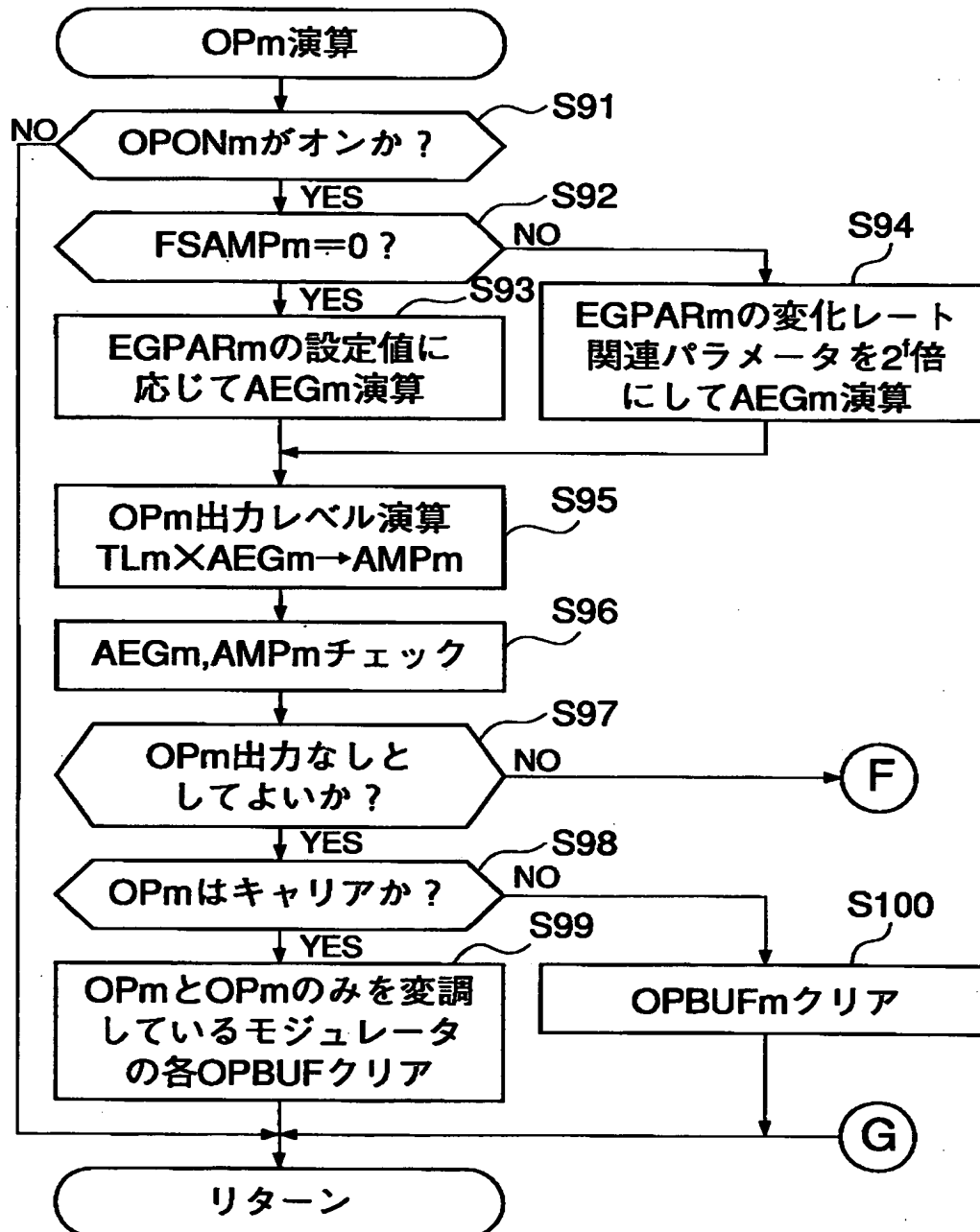
【図15】



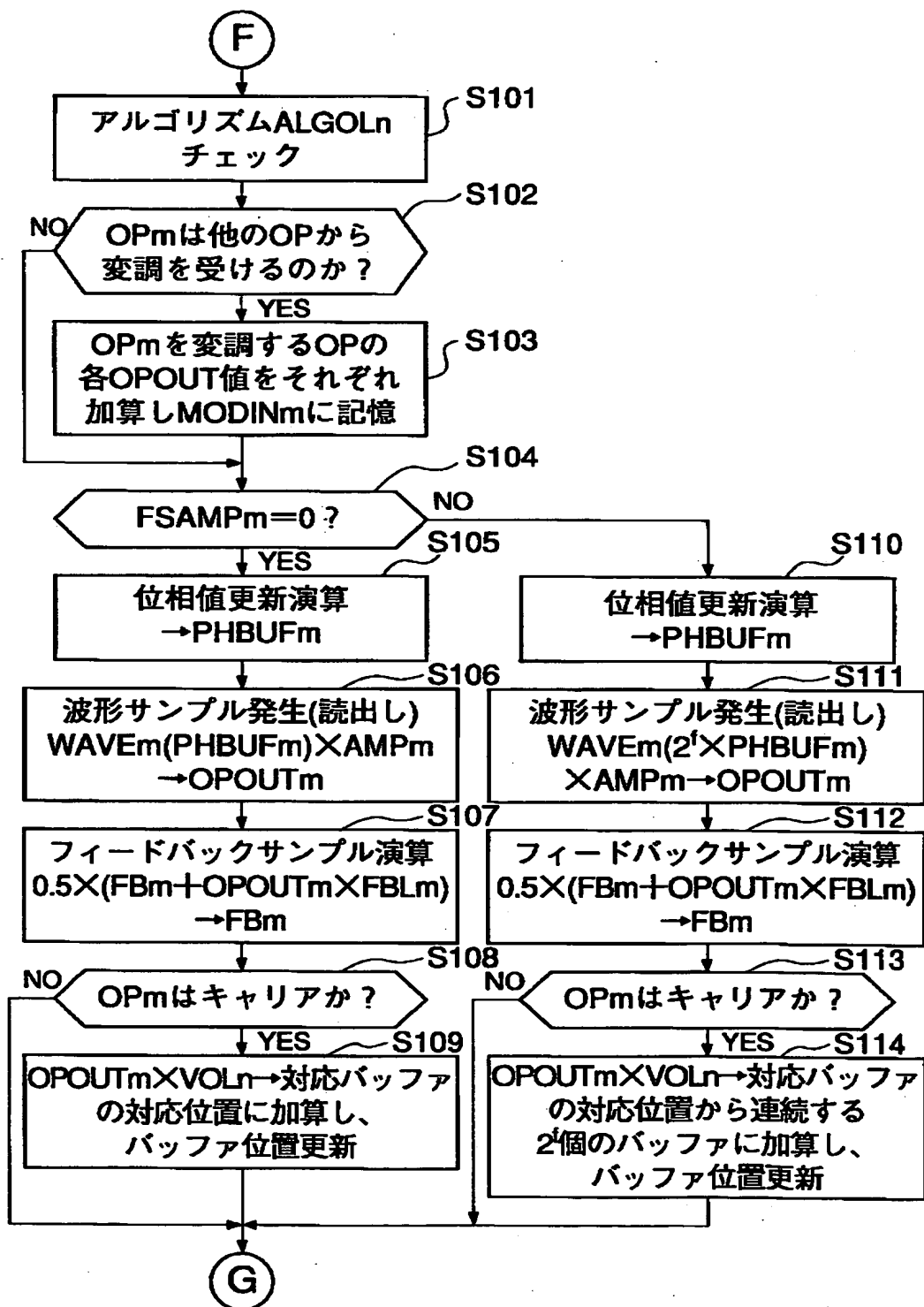
【図16】



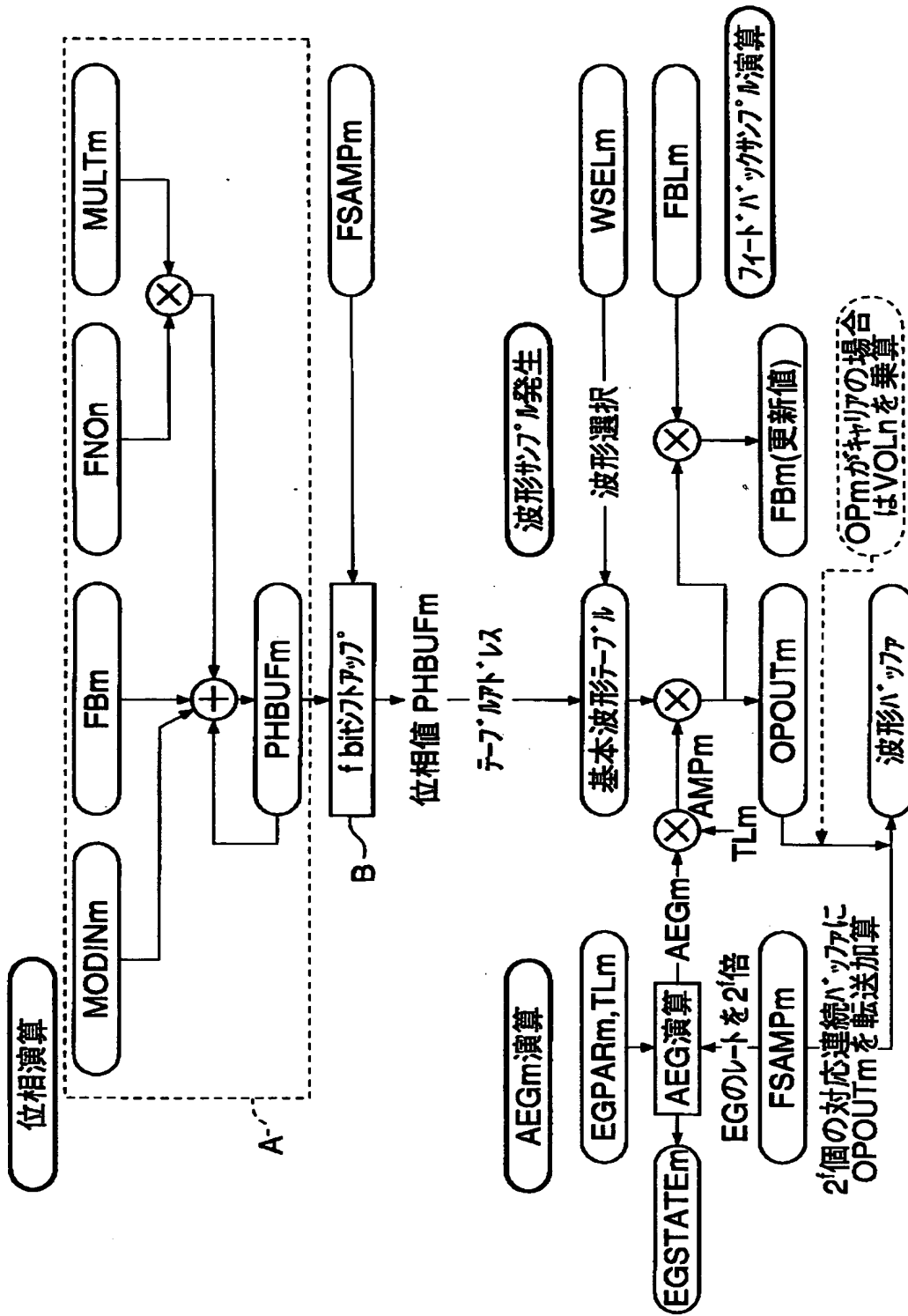
【図17】



【図18】

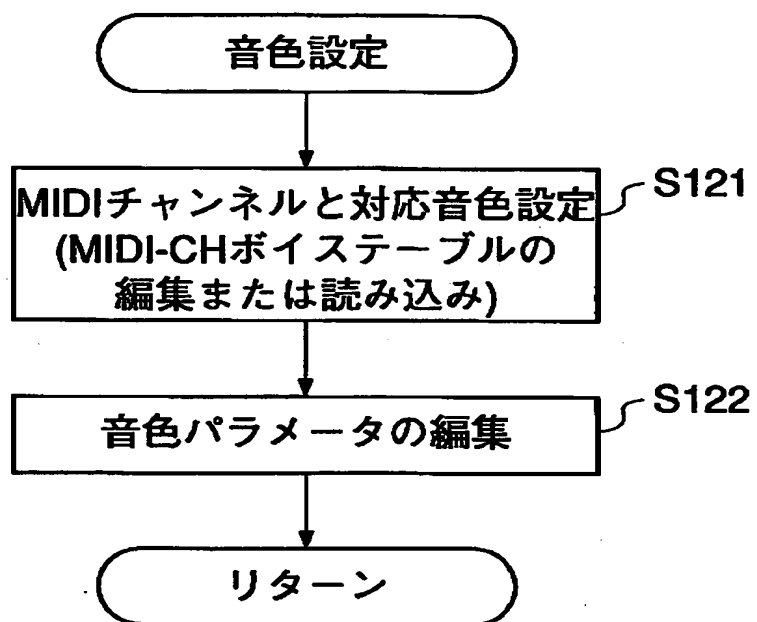


【図19】





【図20】



【書類名】 要約書

【要約】

【課題】 特別な専用ハードウェアを追加せずにソフトウェアプログラムで楽音生成を行うことによりコストを低減化させるとともに、その用途、動作環境、目的および設定等に応じて楽音生成の演算処理に対する演算手段の負荷の軽重および出力楽音波形の品質の高低を変更することが可能なコンピュータソフトウェアを用いた音源システムおよび音響波形データ生成方法を提供する。

【解決手段】 演算対象オペレータ $m$ のサンプリング周波数制定データ $F S A M P m$ の値に応じて振幅制御用 $E G$ を算出する $A E G m$ 演算がなされ（ステップ $S 9 3$ または $S 9 4$ ）、演算対象オペレータ $m$ に対する出力レベル $A M P m$ の演算がなされた（ステップ $S 9 5$ ）後、その演算結果 $A E G m$ 、 $A M P m$ がチェックされ、そのチェック結果により、演算対象オペレータ $m$ からの出力が必要か否かが判別され（ステップ $S 9 7$ ）、必要でないときには、演算対象オペレータ $m$ が「キャリア」であるか否かが判別され（ステップ $S 9 8$ ）、キャリアのときには、演算対象オペレータ $m$ および該オペレータ $m$ のみを変調しているモジュレータの演算が停止される（ステップ $S 9 9$ ）一方、キャリアでなくモジュレータのときには、当該演算対象オペレータ $m$ の演算が停止される（ステップ $S 1 0 0$ ）。

【選択図】 図17

【書類名】 職権訂正データ  
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】 000004075

【住所又は居所】 静岡県浜松市中沢町10番1号

【氏名又は名称】 ヤマハ株式会社

【代理人】 申請人

【識別番号】 100081880

【住所又は居所】 東京都港区虎ノ門1丁目17番1号 虎ノ門5森ビル  
渡部国際特許事務所

【氏名又は名称】 渡部 敏彦

出 願 人 履 歴 情 報

識別番号 [000004075]

1. 変更年月日	1990年 8月22日
[変更理由]	新規登録
住 所	静岡県浜松市中沢町10番1号
氏 名	ヤマハ株式会社